המחשב האלק

עיבוד נתונים ותכנות ביי

יחיאל האן
יצחק עמיהוד



ספרי לימוד והכשרה במדעי המחשב
הוצאת חוד-עמי בשיתוף המכון לפריון העבודה והייצור

המחשב האלקטרוני

עיבוד נתונים ותכנות בייסיק

ספרים נוספים בהוצאת הוד-עמי

1. המחשב האלקטרוני ועיבוד הנתונים, 1968
   - י' עמיהוד
2. תכנון ותכנות יישומי מחשבים, 1975
   - י' עמיהוד
3. הקלט למחשב, 1978
   - ס' מצפי
4. יסודות בניתוח ותכנון מערכות מידע, 1980
   - י' עמיהוד ו-י' קורפל
5. מערכות תקשורת נתונים, 1981
   - א' סקופ
6. ארגון נתונים וניהול קבצים, 1982
   - ר' הייפרמן
7. מיקרו-מחשבים וה-6800, חומרה ותוכנה, 1983
   - א' הראל
8. בסיסי נתונים - עקרונות מודלים ויישומים, 1984
   - ר' הייפרמן

* הספרים באישור משרד החינוך.

תמונת השער באדיבות חברת יבמ ישראל בע"מ
(מקור: IBM Eurocoordination, France, G510-0787-5/83)

המחשב האלקטרוני
עיבוד נתונים ותכנות בייסיק

יחיאל האן
יצחק עמיהוד

ספרי לימוד והכשרה במדעי המחשב
הוצאת הוד-עמי בשיתוף המכון לפריון העבודה והייצור

THE ELECTRONIC COMPUTER

Data Procesing and BASIC Programming

BY

Y. HAHN

I. AMIHUD

הוצאת הוד עמי
לספרי מחשבים בע"מ

ת.ד. 560 רמת-גן 52105
טל-052-541207

הודפס בישראל
ניסן תשמ"ה, אפריל 1985

HOD-AMI LTD.
P.O.Box 560 Ramat-Gan
Israel, April 1985

הקדמה

"מחשב הינו רק מכונה, לכן אין בו איכויות של יצור אנוש או כל יצור חי אחר, מבלי שיסייעו לו בכך. מחשב שאין יודעים להפעילו, נכשל ללא אזהרה ותגובותיו מפתיעות, עלול להביא את האדם המשתמש בו לכעס, אכזבה ותסכול. בנסיבות אלו אנו משליכים על המכונה את רגשותינו ומכנים את המחשב רע ועוין. ואילו מצד שני, כשהוא נוח להפעלה, תגובותיו ברורות והוא עומד לשרותנו בכל עת, הרי יחסנו אליו חיובי ואנו רואים בו עוזר נאמן ומושלם.

אותן תכונות אנושיות שאנו מצפים שתהיינה למחשב, אינן חלק טבעי באמנות הנדסת האלקטרוניקה, או אפילו ביסודה של שפת המחשב, אלא פרי תכנון מדוייק, עיצוב טוב וניסויים מתישים.

מטרה חשובה הינה להפוך את המחשב שיתאים יותר לאדם המשתמש בו, בעיקר זה החסר אימון מיותר במחשבים".

דברים אלו הינם מובאה מתוך דבריו של המדען הראשי של חברת יבמ, ד"ר לואיס ברנסקומב, בכנס שנשא בסימפוזיון הבינלאומי על האדם, המחשב והחברה שנערך בטכניון בשנת 1981.

הספר שלפנינו מכוון לתרום, ולו אך מעט, להשגת המטרה אשר הציב ד"ר ברנסקומב. "המחשב האלקטרוני" נכתב בניסיון להאיר את האספקטים החשובים בהפעלה נכונה ובניצול נבון של הציוד מחד-גיסא, ולתאר את סוד פעולתו של המחשב האלקטרוני כמערכת עיבוד נתונים מאידך גיסא.

ספר זה מיועד במידה שווה לכל אלה אשר צריכים ללמוד את תורת המחשב כדי להשתמש בו למילוי תפקידם, או שהם מכשירים עצמם להשתמש בו בעתיד. אלו הם כל שוחרי דעת, מנהלים, מינהלנים, מהנדסים, סטודנטים ותלמידים.

תודתנו נתונה לחברת יבמ ישראל בע"מ אשר באדיבותה השתמשנו במחשבים שבמרכז החישובים בתל-אביב לעריכת הספר ולהדפסתו.

העריכה נעשתה באמצעות תמ"י (DCF) מעבד המסמכים בעברית, וההדפסה נעשתה במדפסת הלייזר 3800.

התמונות בספר מוצגות באדיבות החברות יבמ ישראל, י.א. מיטווך (נציגת NCR), דיגיטל, HP.

תבואנה על הברכה גם הכתבנית המסורה שלנו, יונת קרן והמגיהה הלשונית טובית הספיל.

רשימת תמונות באדיבות חברות המחשבים

2-1	יבמ
2-2	יבמ
2-3	יבמ
2-4	דיגיטל
3-2	יבמ
3-7	יבמ
3-9	נסיר
4-4	יבמ
4-5.1	דיגיטל
4-6	יבמ
4-10	יבמ
4-14	יבמ
4-15	יבמ
4-17	נסיר
4-18.1	יבמ
4-20	יבמ
5-12	דיגיטל
8-4	בורוז
11-3	יבמ
11-4	נסיר
11-8	יבמ
12-1	HP
12-3	יבמ
12-4	נסיר
12-5	HP

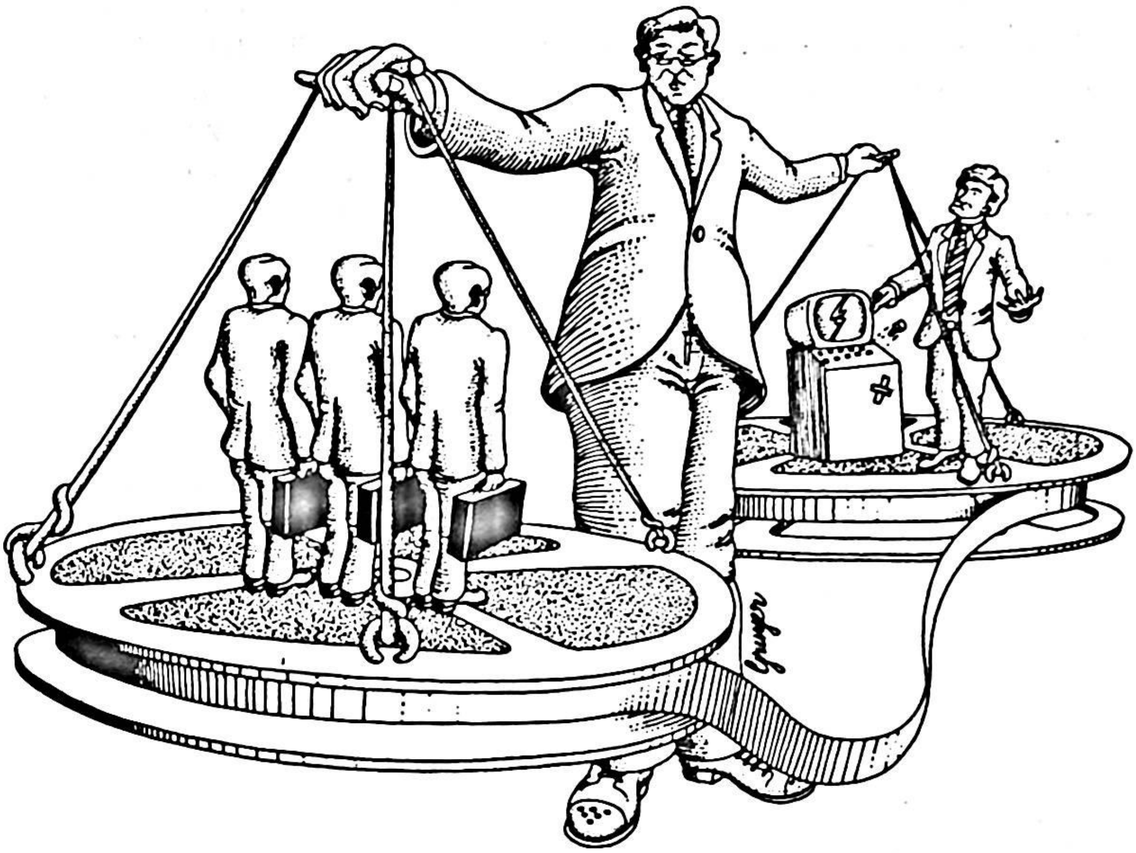
תוכן העניינים

פרק 4 - יחידות היקפיות למחשב 69

פרק 5 - עקרונות התכנון והתכנות 101

פרק 6 - טכניקות תכנות 123

פרק 7 - שפות תכנות 145



פרק 1 – מהחשבוניה עד המחשב

בסוף שנות ה-60 התקיימה תערוכה גדולה בארה"ב ובה הציגו, בין השאר, רובוט המופעל על ידי מחשב. מבקרי התערוכה עמדו בתור לפני ביתן קטן ממדים ודפי נייר בידיהם. על הדף היה הסבר קצר בזו הלשון:

"בביתן נמצא מחשב, אשר מפעיל רובוט. רובוט זה יודע לומר מספר משפטים בשפה האנגלית, הוא יכול לבצע 10 פעולות שונות, שאנו בני האדם, מבצעים. ניתן לבקש ממנו עיפרון או עיתון וחפצים נוספים, אשר מונחים על המדף. נא לשהות בחדר לא יותר מ-60 שניות ולצאת בדלת האחורית. כדי להיכנס לחדר יש להקיש 3 נקישות על הדלת. תודה !"

מבקר אשר הקיש בדלת, ראה, שהדלת נפתחת על ידי גוף מתכת עשוי בדמות אדם. לדמות ראש פח, שתי נורות במקום עיניים וצינורות במקום ידיים ורגליים. - קול מתכתי אמר: "שלום, נא להיכנס!". במרכז החדר היה כיסא, שעליו ישב המבקר ויכול היה לבקש סיגרייה, עיתון ושמונה חפצים נוספים אחרים. כאשר הבקשה היתה "עיתון", ניגש הרובוט בצעדים מדודים אל המדף, אחז בעיתון בעזרת ידיו דמויות המלקחיים והגיש את העיתון למבקר, שישב על הכיסא.

כאשר קם המבקר מן הכיסא, הסתובב הרובוט, ניגש אל הדלת האחורית, פתח אותה לרווחה ואמר: "תודה על הביקור, להתראות!" וכאשר יצא המבקר - סגר הרובוט את הדלת.

בזאת לא תמה הפרשה. מחוץ לביתן חיכו למבקרים עובדי אחד המכונים לחקר דעת הקהל, ושאלו אותם על רשמי הביקור בביתן המחשב. כאשר מוינו התשובות, התקבלה תמונה מעניינת: המבקרים התפלגו לשני מחנות - המחייבים והשוללים.

השוללים אמרו בערך כך: "הביקור איכזב. ציפינו למשהו יוצא דופן, אך מה שראינו, לא הפתיע אותנו. הכל "טריקים" של אלקטרוניקה. הרובוט "דיבר" הודות לרשמקול, שהוסתר בבטנו. הוא ידע לפתוח את הדלת האחורית ולומר מלות פרידה, שעה שקמנו מהכיסא, כי בקומנו, חצינו קרן אור, זו הפעילה מנגנון מסויים, אשר הביא את הרובוט לפתיחת הדלת ולאמירת משפט הפרידה. אלו רק משחקי אלקטרוניקה, שאין בהם כל חידוש".

המחייבים דיברו אחרת: "כמה מעניין, כמה יפה! עכשיו, בסוף שנות השישים מסוגל רובוט זה לבצע כעשר פעולות שונות, כפי שהאדם עושה ואף ביתר אדיבות. הוא אדיב יותר. עד אמצע שנות

השבעים הוא ידע, וודאי, לבצע כ-100 פעולות ולקראת שנות ה-80 אולי אלף פעולות, בדיוק כמונו. עד שנגיע לשנת 1990 יהיה עולמנו מלא מחשבים ורובוטים, שיעשו במקומנו את כל העבודה".

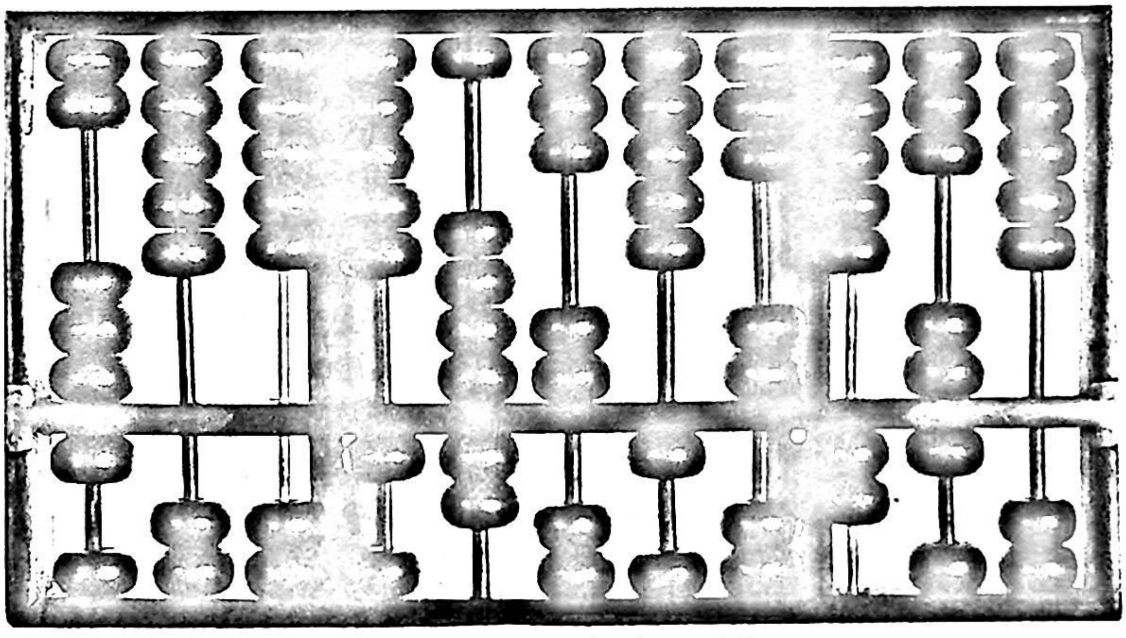
האם ישנם היום, בשנת 1985 רובוטים בעולם? ואם כן, האם באמת מחשב מפעיל אותם? מהו המחשב הזה? כיצד הוא פועל? האם מייצרים מחשבים במדינת ישראל? על שאלות אלה ואחרות ננסה לענות בפרקים השונים של ספר זה.

המחשב האלקטרוני הראשון הופיע בשנות הארבעים של המאה שלנו. אז הוא שימש קבוצה קטנה של אנשים, בעיקר מדענים, אשר ראו בו קודם בראש ובראשונה אמצעי חישוב מהיר ונוח. כיום מסייע המחשב לפתור בעיות בכל תחומי החיים, ועל כך ידובר בהמשך.

אולם אל נא נשכח, שכבר לאדם הקדמון היו בעיות חישוב. יש הטוענים, שפיתוח שיטה של ספירה קדם אפילו להמצאת הכתב. האדם צריך היה לספור את הכבשים שברשותו, רצה לדעת מה מלאי החיטה, הגרעינים, ובכלל - מה רכושו. האדם הפרימיטיבי עשה רשימות על הקירות בתוך מערות, ובעזרת קווים ישרים סימל את היקף רכושו.

1.1 מכונות החישוב הראשונות

לפני למעלה מאלפיים שנה הופיע הכלי המכני הראשון, שבעזרתו יכול היה האדם לעשות את חישוביו. כלי זה נקרא חשבונייה, נהוג לראות בכלי זה את אבי אבות המחשבים שלנו (ראה ציור 1-1).



ציור 1-1 - חשבונייה

מעניין לציין, שהחשבונייה, למרות "גילה" המופלג, עדיין משמשת את ילדינו בגני הילדים ולעיתים נתקלים בחשבונייה אצל עולים, בעיקר ממזרח אירופה ומרוסיה.

באחת מהאוניברסיטאות בארה"ב ניסו לבדוק, מהי תרומת החשבונייה לילדים בגיל הרך. נבחרו 20 גני-ילדים של גילאי 3 לערך וחילקו אותם לשתי קבוצות. ילדים משתי הקבוצות נתבקשו לספור עד 10 ואח"כ נשאלו שאלות כגון: כמה הן שתיים ועוד שלוש? רב הילדים לא יכלו להשיב תשובה נכונה. חלקם אפילו שאלו: שתיים-מה? שלוש-מה? לאחר מכן הוכנס כל ילד לחדר קטן ובו שני שולחנות. על אחד מהשולחנות היו שני כדורים, ועל השולחן השני היו שלושה כדורים. כמעט כל הילדים השיבו תשובה נכונה לשאלה: כמה כדורים יש לנו בסה"כ?

הוחלט בהסכמת הורי הילדים לערוך ניסוי, אשר נמשך שנתיים. בקבוצה הראשונה השתמשו הגננות בחשבונייה ככלי משחק ולימוד שעה אחת ביום. בקבוצה השנייה לא היתה חשבונייה כלל, ולא הושם דגש על עולם המספרים. כעבור שנתיים נבדקו הילדים בידיעת חשבון (ארבע פעולות יסוד, מספרים שלמים בלבד, עד 20). בקבוצה הראשונה הצליחו 85% במבחן ובקבוצה השנייה - פחות מ-50%. על הניסוי הזה חזרו פעמים נוספות והתקבלו תוצאות דומות. המסקנה היתה ברורה: החשבונייה הינה כלי עזר מצויין בהוראת החשבון.

החשבונייה היא כלי בעל כמה תכונות יסודיות חשובות, אשר מעמידות אותה בשורה אחת עם המצאות גדולות אחרות של ימי קדם:

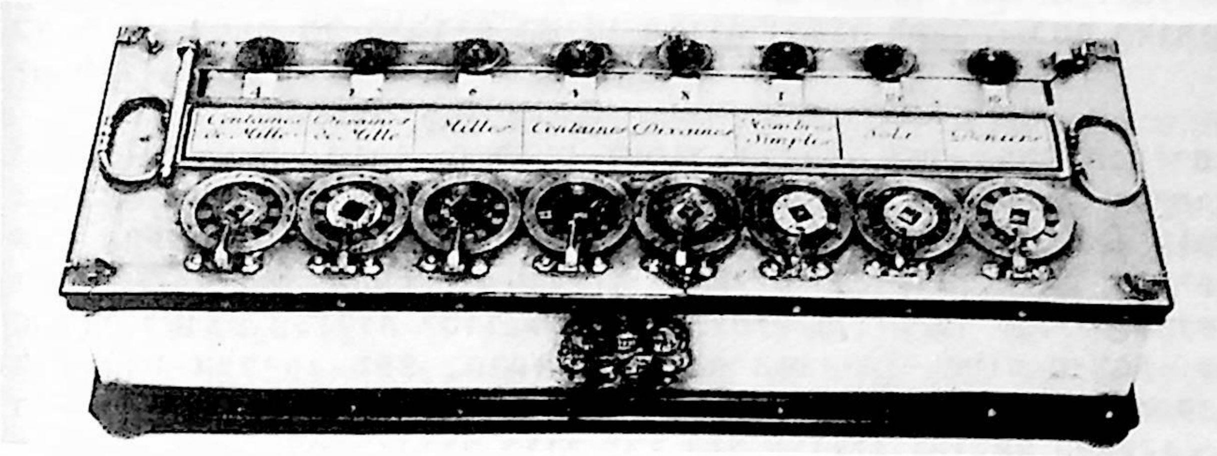
- החשבונייה הינה בעלת <u>**יחידות בודדות רבות**</u>, מעבר לעשר האצבעות, ששימשו את האדם למטרת "חישוב" בראשית ימי ההיסטורייה.

- החשבונייה <u>**"ניטרלית"**</u>. ניתן לספור בעזרתה גרעינים, שקים, בעלי-חיים, פרות, כבשים, וכל דבר אחר.

- החשבונייה <u>**"זוכרת"**</u>. ניתן לסדר את חרוזיה ולהסיט אותם לכיוון מסוים, וכעבור יום, שבוע או חודש לא ישתנה דבר, החשבונייה "תזכור" את המספר (או המספרים), שסודרו קודם לכן. "הזיכרון" הוא מצב החרוזים בחשבונייה ללא כל רישום, כי אנו עומדים לפני תקופת המצאת הכתב.

החשבונייה שרתה את האדם מאות בשנים ללא כל שינוי. באמצע המאה ה-17 המציא בליים פסקאל (BLAISE PASCAL), מתמטיקאי צרפתי, מנגנון מתוחכם, אשר היה מבוסס על שורת גלגלי שיניים בעלי ידית, שנעו על צירים סמוכים זה לזה. זהו למעשה "מיכון" של החשבונייה. כאשר סובבו את הידית של הגלגל הימני ביותר בכיוון השעון, הופיעו הספרות 2,1,0 וכו' בתוך חלון קטן מעל הידית. כאשר נגמר סיבוב שלם הגיעה הספרה 9 ואם המשיכו לסובב את הגלגל, הופיעה שוב הספרה אפס, אך בגלגל שמשמאלו הופיעה הספרה 1. כלומר הגלגל הימני ביותר ייצג את ספרת היחידות, הגלגל שמשמאלו ייצג את ספרת העשרות, ואח"כ את ספרת המאות וכו'. ראה ציור 1-2.

וילהלם לייבניק (G.WILHELM LEIBNITZ), מתמטיקאי והוגה-דעות גרמני, היה מעוניין גם הוא בפיתוח מכונת חישוב. "זו בושה, שאנשים חכמים יבזבזו שעות רבות על "עבודה שחורה" של עשיית חישובים פשוטים", אמר. במשך כ-20 שנה שקד על פיתוח מכונת חישוב וב-1694 יכול היה להפעיל את המכשיר שפיתח, אשר יכול היה לבצע פעולות של כפל וחילוק.



ציור 1-2 - מכונת החישוב של פסקאל

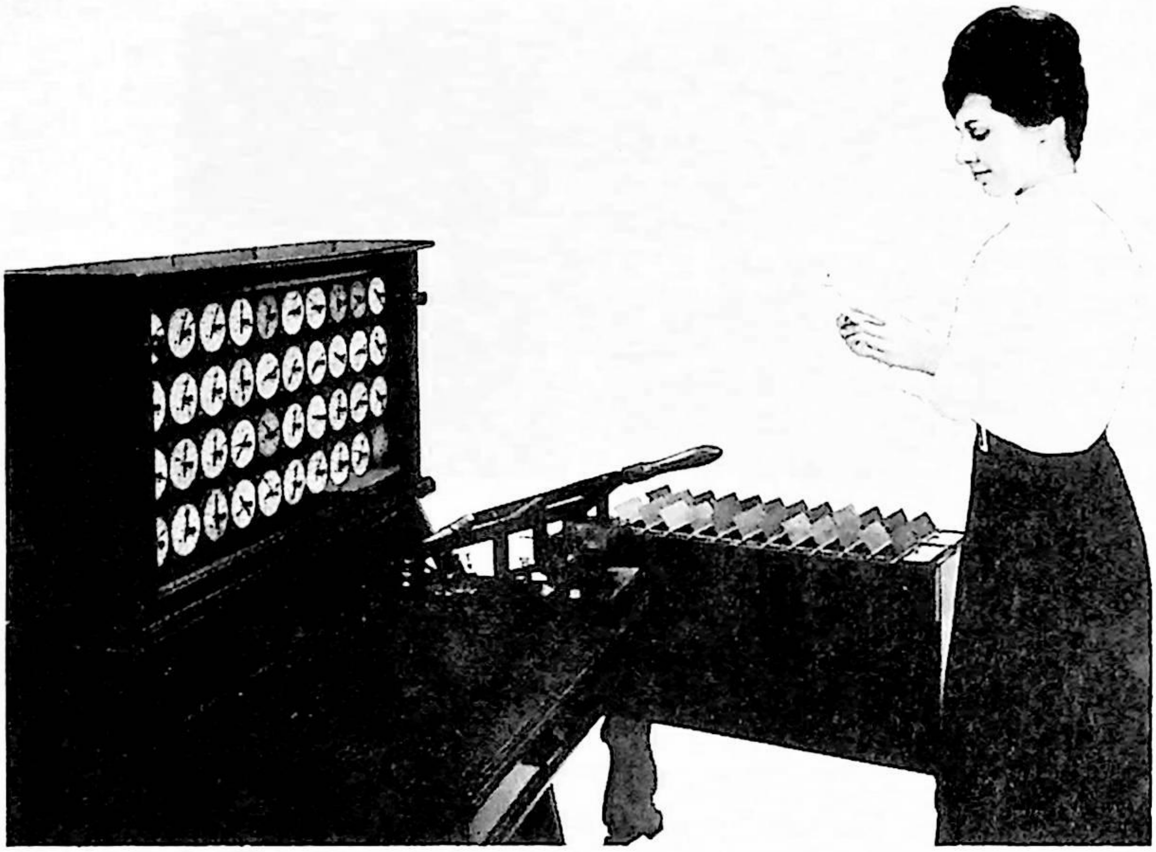
צעד נוסף בהתפתחות מכונות החישוב נרשם לזכותו של מתמטיקאי אנגלי בשם צ'רלס בבאג' (CHARLES BABBAGE). במחצית הראשונה של המאה ה-19 הוא בנה מכונת חישוב מורכבת מאד, שהיתה דומה במבנה שלה למחשבים של ימינו. המכונה הופעלה בעזרת קיטור, כללה יחידת חישוב ויחידת זיכרון. בניית המכונה לא הושלמה, והדגם שלה מוצג כיום במוזיאון המדע במרכז לונדון.

1.2 מכונות ראשונות לעיבוד נתונים

לקראת סוף המאה ה-19 חל אירוע חשוב, אשר גרם לתאוצה גדולה בפיתוח מכונות חישוב אוטומטיות. במפקד האוכלוסין, אשר התקיים בארה"ב בשנת 1880, נאספו פרטים רבים על האוכלוסיה, ועבודת המיון של החומר והחישובים הסטטיסטיים השונים נמשכה שנים אחדות. לקראת 1890, לעת המפקד הבא, חזו גידול של 25% באוכלוסיה, והפוקדים לא האמינו, שיוכלו להתגבר על האוסף העצום של הנתונים והחישובים השונים הכרוכים בעיבוד החומר.

ד"ר הולריית (DR. H. HOLLERITH), אחד מהמנהלים של המפקד, הגה רעיון כיצד להתגבר על הבעייה. הוא החליט להציג את הפרטים החשובים של כל נפקד (כגון שם, שנת לידה, מקום לידה, מצב משפחתי, מקצוע ועוד) בשיטת נקבים בכרטיסים עשוי קרטון. הכרטיסים המנוקבים הוכנסו לשם עיבוד למכונה חשמלית, שהיתה מצויידת במחוונים ("שעונים") רבים (ראה ציור 1-3). את הרעיון להשתמש בכרטיסים מנוקב לשם מתן הוראות למכונות לקח

הולריית ממכונות הז'יקארד (JACQUARD). הז'יקארד היא מכונת אריגה, אשר פעולותיה מודרכות באמצעות סרט קרטון שבו נקובים חורים. הצירוף של הנקבים מפעיל יחידות שונות של המכונה לשם יצירה של דוגמאות האריגה השונות.



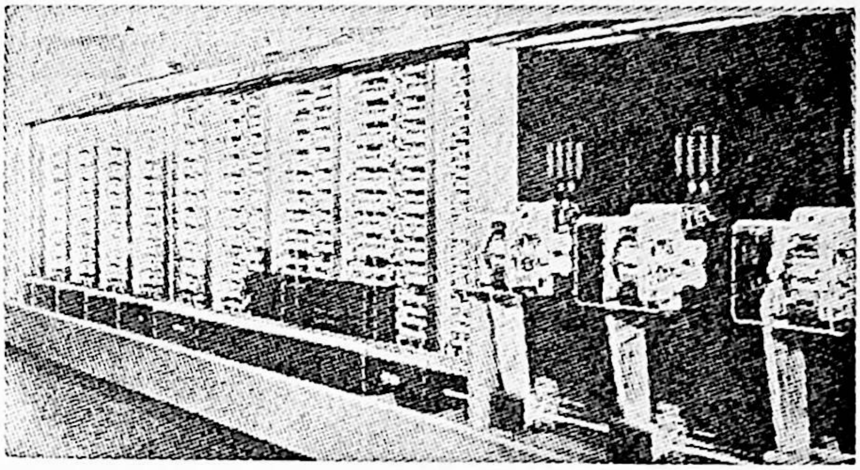
ציור 3-1 - המכונה של הולריית

מפעיל המכונה הניח את הכרטיס על משטח מתכתי וכיוון מערכת של מוטות מתכת אל מול הכרטיס. עם הפעלת המכונה עברו המוטות דרך החורים ונגעו במשטח המתכת, שעליו הונחו הכרטיסים, ובכך סגרו מעגל חשמלי. סגירת המעגל החשמלי גרמה להזזת מחוג על המחוון המתאים בשנתה אחת. ספירה "אלקטרו-מכנית" זו היתה מהירה יחסית ואפשרה לקבל את תוצאות המפקד במחצית הזמן, שנדרש לכך בשיטה, שהיתה נהוגה עד אז.

המכונה של הולריית היתה יכולה בשלב מאוחר יותר למיין את הכרטיסים. המיון נעשה על אחד מהפרטים, שנוקבו בכרטיס, שם, גיל, מקצוע וכדי. לאחר כיוון מוטות המתכת לצירוף המתאים והפעלת המכונה נפלו הכרטיסים המתאימים לתוך תא, שהותקן בתחתית המכונה. פעולת מיון שלימה דרשה סדרה של פעולות בחירה של צירופי מספרים ונמשכה זמן רב.

הרעיון של הכרטיס המנוקב אומץ על-ידי חברת יבמ כבר בתחילת המאה העשרים. מאז היא פיתחה ושכללה את המכונות לקריאת כרטיסים ומיונם, אולם במקביל למכונות של כרטיסים פיתחה חברת יבמ גם מכונות חישוב. מכונת החישוב הבולטת ביותר היתה מארק-1

(MARK-1) - מכונת החישוב הספרתית **האוטומטית** הראשונה. המכונות לעיבוד נתונים באמצעות כרטיסים מנוקבים הופעלו בארץ ובעולם שנים רבות, עד לאמצע שנות ה-60.



ציור 4-1 - MARK-1

בניית המכונה מארק-1 החלה ב-1937 לפי התכניות של הווארד א.אייקן, בעידוד חברת יבמ. היא היתה מכונת חישוב אלקטרומכנית מהירה מאוד וגדולה מאוד. היא שקלה כחמישה טון והיו בה 800,000 מטר של חוטי חשמל. ייחודה של המכונה היה בכך, שניתן היה להזין אותה בתכנית ארוכה, אשר נוקבה בסרט נייר. מרגע שהמכונה הופעלה, היא עבדה באופן <u>עצמאי</u>, עד שסיימה את עבודתה. היא יכלה להכפיל שני מספרים בני 23 ספרות בתוך שש שניות, והיתה מסוגלת לנקב את התוצאה על סרט נייר או להדפיסה על מכונת כתיבה, שחוברה אליה.

מארק-1 פעלה ללא דופי עד 1959 ואז ניתנה במתנה לאוניברסיטת הרווארד, ארה"ב, ומאז היא מוצגת שם.

<u>1.3 המחשב כמכשיר לעיבוד נתונים</u>

נכיר עתה את המחשב, ונבחן מה הוא עושה, וכיצד הוא עושה זאת.

נתחיל בדוגמה מחיי היומיום. כל אחד מאתנו משתמש בחשמל ומדי חודשיים מגיע אל כל אחד מאתנו חשבון, ובו דרישה לתשלום בעבור כמות החשמל, שצרכנו בתקופה החולפת. כיצד מכינים את חשבון החשמל? נראה כיצד עשו זאת לפני השימוש במחשב.

אצל כל צרכן במדינת ישראל מותקן מונה חשמל, הן בבית פרטי והן במבנה תעשייה או במשרד. נציג חברת החשמל מגיע פעם אחת

בחודשיים ורושם את מצב המונה. לצורך דוגמא בלבד נניח, שבאחת מקריאות המונה, הראה המונה את המספר 4569, ובקריאה, שהתבצעה כחודשיים לאחר מכן, הראה המונה את המספר 4851. בחברת החשמל נשמר המספר מן הקריאה הראשונה. פקיד החברה מבצע חיסור: 4851-4569=282. המספר 282 מציין את מספר הקילווטים, שצרכנו בתקופה שבין שתי הקריאות. הפקיד מכפיל את המספר 282 בתעריף המקובל (נניח 10 שקלים לכל קילווט) וכך מתקבל הסכום לתשלום: 2820 שקלים, אשר מודפס בחשבון, שנשלח אל הצרכן. בחשבון החשמל רשומים גם שם הצרכן, כתובתו, מספר המונה ופרטים נוספים.

נבחן את תהליך הפקת החשבון לחלקיו. בראש ובראשונה נאספו כל הפרטים הדרושים לחישובים. פרטים אלה נקראים בשם כולל <u>נתונים</u> (DATA). נתונים הם ביטוי לעובדות ולמצבים מסויימים, שעשויים להיות בעלי משמעות שימושית למקבל אותם. נתונים יכולים להיות מספריים, כגון מספר קילווטים, מחיר לקילווט, מספר צרכן, מספר תעודת-זהות וכדומה. ישנם נתונים, שאינם מספריים כגון שמו של אדם, כתובתו, צבע שערותיו וכו'.

הנתונים משמשים חומר גלם לתהליך שלם, שנקרא <u>עיבוד נתונים</u> (DATA PROCESSING). בדוגמה שלנו פעולות החישוב השונות מהוות את עיבוד הנתונים. במקרה שלפנינו חישוב הצריכה בקילווטים והכפלתה בתעריף - הן פעולות של עיבוד נתונים. התוצאה של עיבוד הנתונים היא <u>המידע</u> (INFORMATION) ובמקרה שלנו המידע הינו חשבון החשמל המודפס. המידע נדרש לשם ביצוע פעולה כלשהי, למשל תשלום בעבור צריכת חשמל, או לשם קבלת החלטה כמו בחירה בין רכישת שני מוצרים על פי מחירם. עיבוד הנתונים הינו תהליך, שתפקידו להפוך את הנתונים (חומר גלם) למידע (מוצר מוגמר).

תהליך עיבוד הנתונים כולל פעולות מסוגים שונים:

- פעולות אריתמטיות.
- בדיקות שונות של מהימנות ושלימות הנתונים.
- פעולות לוגיות הקשורות בקבלת החלטות שונות שאינן בהכרח כמותיות.

הפעולות הללו, כאשר הן מסודרות ברצף הגיוני כדי לפתור בעייה מסויימת מהוות <u>תכנית</u> (PROGRAM). כפי שהפקיד בחברת החשמל ביצע פעולות שונות בזו אחר זו, שבסופן קיבל את הסכום לחיוב הצרכן, כך ניתן ליצור תכנית למחשב, אשר תבצע את עיבוד הנתונים.

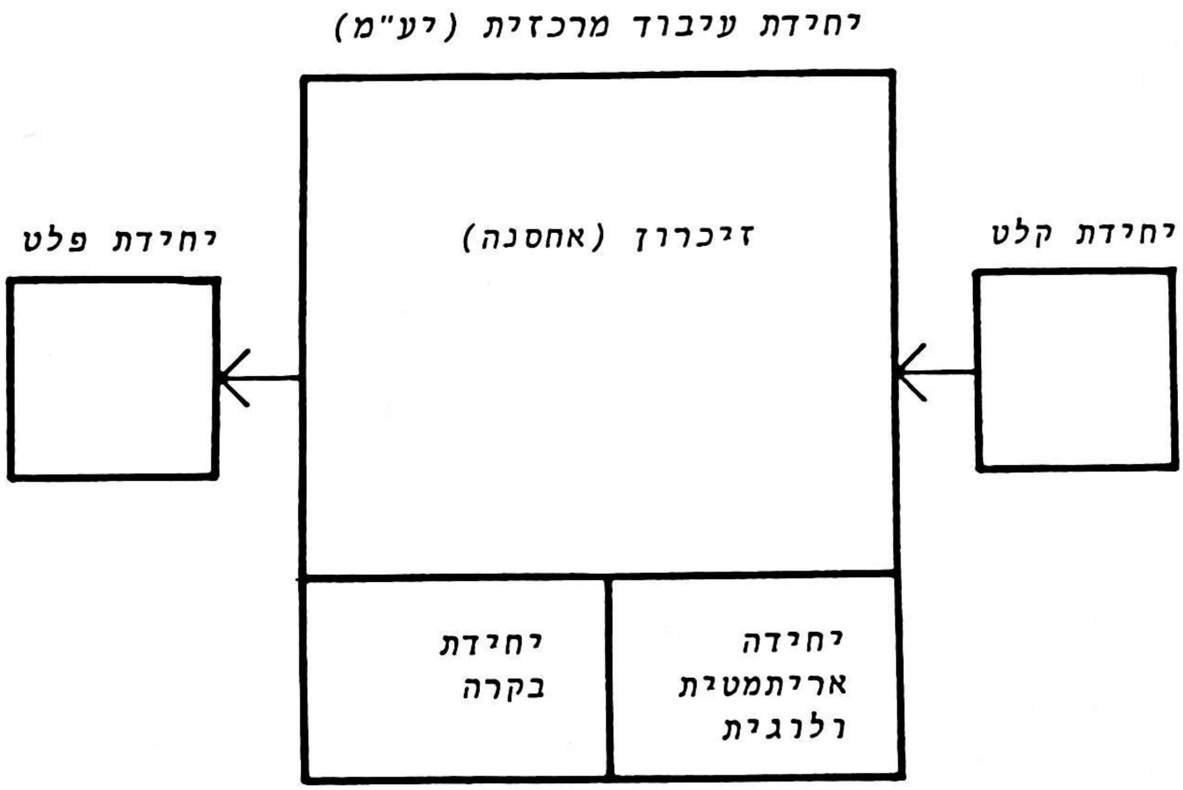
הבה נחזור אל חשבון צריכת החשמל וננתח אילו סוגים של פעולות בוצעו בזמן הכנתו. השלב הראשון בפעילות היה איסוף הנתונים והכנתם לעבודה. שלב זה נקרא שלב <u>הקלט</u> (INPUT). סדרת פעולות אחרת היתה קשורה בחישובים שונים, שהם שלב <u>העיבוד</u> (PROCESSING). השלב האחרון בדוגמא היה שלב הדפסת החשבון,

שהוא <u>הפלט</u> (OUTPUT). הסדרה השלימה של כל הפעולות, החל מאותו רגע, שבו הפקיד קרא את מספר הצרכן, והתחיל בחישוב, ועד הרגע, שבו הודפס החשבון הסופי, הינה למעשה <u>תכנית עבודה.</u> לפי תכנית זו, על הפקיד לחשב עבור כל צרכן וצרכן, מהו הסכום, שמגיע לחברת החשמל בעבור הצריכה של אותה תקופה.

כמעט בכל הפעולות של עיבוד נתונים ניתן להבחין בשלושת השלבים <u>קלט-עיבוד-פלט.</u>

הכנת חשבון החשמל מבוצעת בחברת החשמל זה עשרות שנים, ועד להפעלת המחשב הראשון בחברה היא נעשתה כפי שתיארנו קודם.

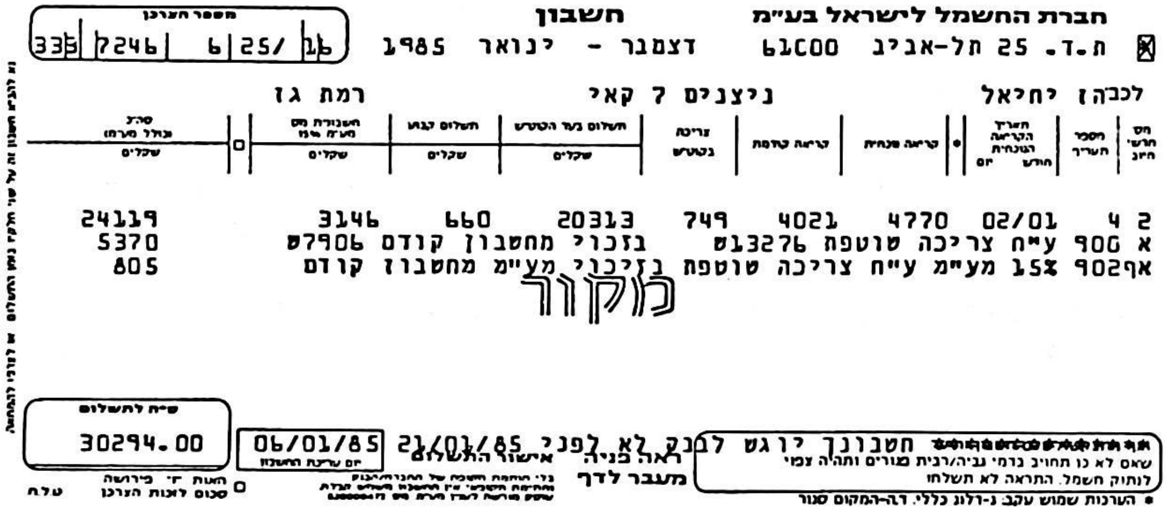
כאשר נרכש המחשב הראשון של חברת החשמל היתה כוונה להשתמש בו לשם עיבוד נתונים. בוני המחשבים הראשונים תכננו אותו <u>כמכשיר לעיבוד נתונים</u> והתקינו בו חלקים, שעוסקים בפעולות <u>קלט,</u> חלקים אחרים, שעוסקים בפעולת <u>עיבוד,</u> וחלקים נוספים שמבצעים את פעולת <u>הפלט.</u> בכל מחשב ישנם <u>התקני קלט</u> (INPUT DEVICES), <u>התקני פלט</u> (OUTPUT DEVICES) <u>ויחידת עיבוד מרכזית</u> (CENTRAL PROCESSING UNIT). נוהגים לכנות את הצירוף של שלושת סוגי היחידות <u>מערכת אלקטרונית לעיבוד נתונים</u> (ELECTRONIC DATA PROCESSING SYSTEM) או <u>מערכת מחשב</u> (COMPUTER SYSTEM).



ציור 5-1 - מערכת מחשב סכימטית

חברת החשמל מבצעת כיום את כל הפעולות של הפקת חשבון חשמל בעזרת מערכת המחשב שלה. הנתונים שנאספו על ידי נציגי החברה,

מוזנים למחשב באמצעות יחידת קלט (למשל קורא אופטי). הנתונים של הפעם הקודמת, שנשמרו על סרט מגנטי, מוזנים גם הם אל המחשב. ביחידת העיבוד המרכזית מתבצעות פעולות החישוב בעזרת תכנית, אשר הוזנה קודם לכן אל המחשב. בגמר החישובים מודפס חיוב צריכת החשמל במדפסת (יחידת פלט). הקריאה החדשה נכתבת על סרט מגנטי, כדי לאפשר עריכת חשבון בחודש הבא.



ציור 1-6 - טופס חיוב צריכת חשמל

נתבונן בטופס חיוב צריכת החשמל שבציור 1-6. רשומים בו פרטים רבים, קבועים מראש, המאוחסנים בסרטים מגנטיים, כדי לאפשר הכנת חשבון כזה גם בחודשים הבאים. הפרטים הללו מהווים את הנתונים, שהם חומר הגלם לעיבוד הבא.

כל נתון בטופס נקרא שדה (FIELD). שדה הינו הנתון הבסיסי הקטן ביותר, שיש לו משמעות. לדוגמה: מספר צרכן, שם צרכן, קריאה נוכחית וכו'. שדה מורכב מתווים. תו (CHARACTER) הינו שם כולל לכל הסימנים, שמרכיבים שדה כמו אות, ספרה, או סימן כלשהו.

השדות מתחלקים, מבחינת התוכן, לשלושה סוגים:

- שדה מספרי (NUMERIC FIELD) הינו שדה המורכב מספרות בלבד וניתן לבצע בו פעולות חשבון. לדוגמה: מחיר, כמות וכד'.

- שדה אלפא-ביתי (ALPHABETIC FIELD) יכיל רק את הסימנים של האלף-בית העברי (או ה-ABC של שפה זרה כלשהי), את הסימן רווח (SPACE או BLANK) בין המלים, וכמה סימנים נוספים מוסכמים.

- שדה אלפא-נומרי (ALPHANUMERIC FIELD) מוגדר כשדה, שיכול להכיל את כל התווים המותרים.

אוסף כל השדות השייכים לאותו פרט נקרא **רשומה** (RECORD). רשומה מוגדרת כאוסף שדות, השייכים לאותו פרט או פריט באוכלוסייה מסויימת. לדוגמה, רשומת פריט במחסן תכיל את השדות הבאים: מס' קטלוגי, שם פריט, מחיר ליחידה, שם יצרן וכו'.
רשומת תלמיד בבית-ספר תכיל כמובן שדות אחרים: שם התלמיד, מס' ת.ז., שם האב, כתובת וכו'.

אוסף הרשומות של אוכלוסייה מסויימת נקרא **קובץ** (FILE). למשל, קובץ הסטודנטים באוניברסיטה יכיל את כל הרישומים על הסטודנטים. קובץ הצרכנים בחברת החשמל מכיל את כל הרשומות של הצרכנים, שאצלם מותקן מונה חשמל.

יש לציין, שאין מגבלה כמותית לגבי מספר המופעים של שדות, רשומות או קבצים. כל רשומה יכולה להכיל מספר בלתי מוגבל של שדות (מובן שאין זה מעשי, שרשומה תכלול אלפים ורבבות של שדות!). קובץ המוגדר לאוכלוסייה מסויימת יכול להכיל רשומות בכמות גדולה מאוד. קובץ התושבים של סין העממית יכיל כמיליארד רשומות!. הנושא של ארגון קבצים יטופל בהרחבה בפרק 9 של ספר זה.

1.4 מחשבים אלקטרוניים - דורות שונים

המאורעות הדרמטיים של מלחמת העולם השנייה יצרו ביקוש הולך וגדל לעיבוד נתונים בכמויות עצומות ובבהירות רבה. הן אנשי הצבא והן אנשי הכלכלה הדגישו את הצורך הדחוף במכונות חישוב מהירות יותר. החל מירוץ בין האוניברסיטאות בארה"ב ובאירופה (בעיקר אנגליה) לפיתוח מכונות חישוב, שתהיינה מהירות הרבה יותר מכל מכונה אחרת. המדענים פנו אל התחום המהיר ביותר, שהיה מוכר להם - החשמל.

הפולס החשמלי נע במהירות של קרן אור שהיא 300,000 ק"מ בשנייה. אם היינו מצליחים להדליק נורת חשמל בעצמה גדולה מאוד עד כדי כך שקרנה תגיע אל הירח, אז היתה קרן האור מן הנורה שלנו עושה דרכה אל הירח במעט יותר משנייה, כי המרחק מכדור הארץ לירח הוא 384,000 ק"מ. נמחיש מהירות זו בדוגמא נוספת: אורך קו המשווה הוא כ-40,000 ק"מ. אות (SIGNAL) חשמלי הנשלח מנקודה מסויימת על קו המשווה יכול להקיף את כדור הארץ שבע פעמים במשך שנייה אחת! החשמל מביא תועלת לאדם, החל ממכונות ונורות חזקות מאוד (בהם זרימת החשמל נמדדת בעשרות ומאות אמפרים) ועד לרדיו הביתי, שבמעגליו הזרימה חלשה (היא נמדדת באלפיות האמפר או אפילו בפחות מזה).

בוני המחשבים הראשונים פנו אל המעגלים האלקטרוניים, אשר דומים למעגלי הרדיו. הם החליטו להשתמש במרכיבים של מעגל אלקטרוני: שפופרת הריק, קבלים, נגדים, סלילים וכו' ונמנעו

שימוש בחלקים מכניים נעים, כגון גלגלים, מונים וגלגלי שיניים. מאז שהותקן המחשב הראשון, חלה התפתחות רבה בתחום המחשבים במקביל להתקדמות באלקטרוניקה ובמחקר המדעי בנושאי המחשוב. מקובל לדבר על "דורות מחשבים", אשר כל אחד מהם מציין שלב בהתפתחות מערכות המחשבים ובשימושים שלהם.

1. דור ראשון של מחשבים (1946-1958)

המחשב אשר מוכר כמחשב האלקטרוני הראשון הוא אניאק (ENIAC- ELECTRONIC NUMERICAL INTEGRATOR AND CALCULATOR) אשר נבנה באוניברסיטת פנסילבנייה עבור צבא ארה"ב. פיתוחו קשור בשני שמות: פרספר אקרט וד"ר ג'והן מאויקלי. הם השתמשו במעגלים האלקטרוניים, שבהם "הדליקו וכיבו" מנורות ריק. מנורה דלוקה יצגה את הספרה 1 ומנורה כבויה - את הספרה 0. בכך הניחו את היסוד לשימוש בשיטה הבינרית בעולם המחשבים (על השיטה הבינרית ראה בהמשך, בפרק 3). סמוך להשלמת הבנייה של המחשב הראשון בשנת 1945 הציע ד"ר ג'והן פון נוימן, אחד מאבות הקיברנטיקה, שבוני המחשב ידאגו לכך, שגם הנתונים וגם התכנית, שמעבדת אותם יאוחסנו בתוך יחידה מיוחדת של המחשב, שקרויה זיכרון (MEMORY).

רעיון התכנית המאוחסנת, שהוצע ע"י ד"ר ון-נוימן (DR. VON NEUMAN) והשימוש במעגלים אלקטרוניים הם, שמבדילים באופן ברור בין מכונות החישוב השונות לבין המחשבים האלקטרוניים, שהופיעו סמוך לאמצע המאה. מחשבים נוספים ראויים לציון הם: המחשב "האירופאי" הראשון אדסאק (EDSAC), שהחל לפעול באוניברסיטת קימברידג' אנגלייה בשנת 1949, EDVAC שנבנה ב-M.I.T ו-UNIVAC, שפעל באוניברסיטת פרינסטון בארה"ב.

במקביל לפיתוח המחשבים במוסדות להשכלה גבוהה הוחל בייצורם לשימוש מסחרי. חברת יבמ יצאה בשנת 1952 עם סדרת מחשבים, שנקראה 700. מחשבים אלה היו מהירים מאוד בתקופתם ואפשרו ביצוע מאות פעולות חיבור בשנייה אחת. המחשבים של הדור הראשון היו גדולים ומסורבלים מאוד, כי מעגליהם התבססו על שפופרת הריק בדומה למעגלים של מקלטי הרדיו הישנים. אולם בעוד שבמקלט רדיו של אותם ימים היו 5 או 8 מעגלים כאלה, הרי במחשבים הוכנסו כ-18,000 מעגלים ושפופרות. המחשב הותקן על שטח גדול של עשרות מטרים מרובעים והקרין חום רב מאוד, שפופרות רבות נשרפו ועבודת המחשב השתבשה. אפילו קירור או מיזוג אויר לא הועילו. לסדרה זו של מחשבים שייכים גם E101 של חב' BURROGHS, DATAMATIC 1000 של חברת HONEYWELL.

מחשבים מן הדור הראשון הגיעו גם למדינת ישראל ופעלו בה

שנים רבות. מחשבים אלה אפשרו, למרות חסרונותיהם, לבצע בשנייה אחת חישובים, שקודם לכן נדרשה להם עבודת מהנדס במשך שבוע שלם!

2. <u>דור שני של מחשבים (1958-1964)</u>

הופעת הטרנזיסטורים בשנות החמישים חוללה מהפכה בתעשייה האלקטרונית וכמובן גם בתעשיית המחשבים. הטרנזיסטור הבנוי מחומר מוצק מוליך למחצה, יכול לבצע את הפעולות, שהוטלו בזמנן על שפופרת הריק במהירות ובאמינות גבוהות יותר. הממדים הקטנים אפשרו לייצר מחשבים בעלי נפח קטן יחסית, היות שהפולס החשמלי לא היה צריך לעשות דרך ארוכה כמו קודם. הוגברה מהירות הפעולה של המחשבים הללו והם יכלו לבצע כ-10,000 פעולות חיבור בשנייה אחת (יבמ, סדרות 7000, 1400, בורוז 5000, פילקו 2000, CDC 600 ועוד).

במדינת ישראל פעלו מחשבים בני הדור השני במשרדי ממשלה ובבנקים. הם החליפו את מחשבי הדור הראשון והצטיינו ביציבות ובאמינות. שניים מהם פעלו עד סוף שנות השבעים בשתי חברות ציבוריות בארץ. הפעלת המחשבים האלה בישרה למעשה את התחלת השימוש בעיבוד נתונים סדיר בתחומים שונים של המשק הישראלי.

3. <u>דור שלישי של מחשבים (1964-1971)</u>

בשנת 1964 הופיעו המחשבים הראשונים של הדור השלישי, אשר מתאפיין במספר תכונות חשובות:

א. שימוש בטכנולוגיה חדישה

במעגלי המחשב שולבו <u>שבבים</u> (CHIPS) במקום הטרנזיסטורים של הדור השני. שבבים אלה הינם זעירים מאוד. מסופר, שאחד המדענים ניסה לבדוק כמה שבבים כאלה ניתן להכניס לאצבעון רגיל בו נעזרת תופרת בעבודתה, והתברר לו שכדי למלא אצבעון ממוצע יש צורך בכ-50,000 שבבים כאלה.

שילוב של שיטות כימיות ואופטיות מאפשר <u>הזערה</u> (MINIATURIZATION) של כל המעגל, וכך ניתן להתקין על משטח זעיר עשוי קרמיקה <u>מאות</u> מעגלים של מחשב בטכנולוגייה הנקראת <u>SLT</u> (SOLID LOGIC TECHNOLOGY). מחשבים מן הדור הזה מסוגלים לבצע עד מליונים של פעולות חיבור בשנייה אחת.

ב. הופעת סדרות של מחשבים תואמים ומודולריים

רב יצרני המחשבים הוציאו לשוק "משפחה" של מחשבים, שהיא סידרה, שבה ישנם מחשבים קטנים וגדולים. כל המחשבים פועלים בשיטה דומה ולכן המעבר ממחשב קטן לגדול, כאשר נדרשת עצמת חישוב גדולה יותר, אינו מחייב שינוי בתכניות העיבוד.

ג. הגדלת הזיכרון הפנימי של המחשב

הגדלת הזיכרון למאות אלפים של תאים אפשרה לעבוד בשיטת <u>התכנות המרובב</u> (MULTI-PROGRAMMING), שבאמצעותה ניתן להזין לזיכרון מספר תכניות ולבצע אותן במקביל תחת פיקוח של תכנית מרכזית מיוחדת - <u>מערכת הפעלה</u> (OPERATING SYSTEM). מערכת ההפעלה היא תכנה סטנדרטית, המסופקת על ידי היצרן לשם ניהול משאבי מערכת המחשב והפעלת התכניות השונות ברצף. (נושא זה יוסבר בהרחבה בפרק 8).

ד. עיבודים מקוונים

השיפורים בטכנולוגייה של המחשבים מבחינת החמרה והתכנה סללו את הדרך לשיטות עיבוד חדשות. החשובה בהן היא ביצוע עיבודים <u>מקוונים</u> (ON-LINE) באמצעות <u>מסופים</u> (TERMINALS), המחוברים אל המחשב בקרבתו ואף ממרחקים של עשרות ואלפי קילומטרים.

בישראל נקלטו מאות מחשבים של הדור השלישי. נזכיר את הסדרות הבולטות IBM-360, CDC-6000, BURROUGHS 6500-7500, NCR-CENTURY.

4. <u>דור רביעי של מחשבים (1971-1983)</u>

בתחילת העשור השביעי של המאה שלנו הופיע הדור הרביעי של המחשבים. במה מתאפיין דור זה?

א. תכנון

במערכות מחשבים של דור זה דאגו המתכננים להפריד את יחידות הקלט והפלט מיחידת העיבוד ולהגדיל את יכולת הביצוע שלהם. הפרדה זו והוספת מאגרי זיכרון מאפשרים עיבוד נתונים בכמות גדולה יותר ובמהירות רבה יותר.

ב. טכנולוגייה מתקדמת

השימוש <u>במעגלים מונוליטיים MOS</u>
(METAL-OXID-SEMI CONDUCTOR INTEGRATED CIRCUIT)
מאפשר מזעור היחידות האלקטרוניות וייצור מחשבים בעלי

מליונים רבים של תאי זיכרון ובעלי מהירות ביצוע עד 25,000,000 פעולות חיבור בשנייה.

ג. הרחבת השימוש במערכת מחשב ל"רוחב" ול-"עומק"

השימוש במחשב הולך ומתרחב ומגיע לתחומים חדשים ומגוונים. כמעט שאין שדה בתחום הכלכלה והמנהל, הייצור והחקלאות בהם טרם הופיע המחשב או לפחות מסוף של מחשב.
במלה "עומק" כוונתנו להגברת השימוש במחשב בתוך ארגון או מפעל אחד בכל השכבות והדרגים. ישנם ארגונים, שבהם מופעלים מסופי מחשב באולמות הייצור ובמשרדים, והם עומדים לרשות אנשי מנהלה ומנהלים זוטרים ובכירים וגם לרשות הלקוחות (כמו בבנקים למשל).

ד. השימוש במחשב האישי (PERSONAL COMPUTERS)

לקראת סוף שנות השבעים הופיעו בשוקי העולם מחשבים קטני ממדים, בעלי עצמת חישוב גבוהה ונוחים לשימוש. המחשב הינו דמוי מסוף, שלו לוח קלידים ומצג הכולל גם את יחידת העיבוד. אפשר לחבר לו יחידות קלט/פלט לרישום מגנטי. ישנם מחשבים הכוללים לוח קלידים ויחידת עיבוד וניתן לחבר אותם למקלט טלויזייה ביתי לשם תצוגה. בדרך זו מגיע המחשב מן המשרד גם אל הבית. בין המחשבים הללו ישנם קטנים, שהשימוש בהם מוגבל למספר קטן של תחומי פעולה, אך נמצא ביניהם גם כאלה, שיכולים לבצע את כל הפעולות והעיבודים של מחשב מן הדור השני.

אם יש סימנים של "מהפכה" בתחום של עיבוד הנתונים, הרי היא נישאת על "כתפיהם" של המחשבים האישיים.

5. דור חמישי של מחשבים (-1984)

בסוף חודש מרץ 1984 נערך יום עיון מטעם האיגוד הישראלי לחקר העתיד. בכנס זה השתתפו עתידנים-מדענים, שעיסוקם חקירת ההווה והסקת מסקנות לגבי התפתחויות והתרחשויות בעתיד. אחת ההרצאות נישאה ע"י ד"ר יעקב בעל-שם מאוניברסיטת תל-אביב. דובר בה, בין היתר, על דור חדש של מחשבים - הדור החמישי. מתברר, שבכמה מקומות בעולם, בעיקר ביפן ובארה"ב, פיתוח מחשב זה הגיע לשלב מעשי מאד.

הטכנולוגייה של הדור החמישי מתאפיינת, מבחינת החמרה, בשתי נקודות עיקריות: האחת נקראת <u>VLSI</u> (VERY LARGE SCALE INTEGRATION) ופירושה - מיזעור בקנה מידה חסר תקדים. חברת יבמ הודיעה בחודש יוני 1984 על

שהצליחה לייצר במעבדות הפיתוח שלה שבב (CHIP) שמורכבים בו מליון רכיבים. זוהי קוביה קטנה, שגודלה אינו עולה על 0.5 סמ"ק! הסיבה לכך מובנת - מהירות האור, בה עובדים המעגלים האלקטרוניים של המחשב, אינה ניתנת להגדלה, היא קבועה ולכן יש צורך לקצר את המרחקים אותם יעברו האלקטרונים מרכיב לרכיב.

המחשבים של הדור הבא יהיו מהירים מאד ויבצעו, כנראה, מיליארד פעולות בשנייה. המסופים המחוברים אל המחשב יהיו מסכים בעלי <u>מסך-מגע</u> (TOUCH SCREEN), שעליו יוקרנו נתונים, ציורים או שאלות ותשובות מוכנות. המשתמש יפעיל את המסוף במגע של אצבעות ידיו במקומות הרצויים על פני המסך. בנוסף, יהיו למחשב מנגנוני חישה לקלט מסוגים שונים.

המונח "הדור החמישי" אינו מתיחס לחמרה בלבד. גם בשטח התכנה צפויות התפתחויות. ארבעת הדורות הראשונים של המחשבים התבססו על עקרון התכנית המאוחסנת של ד"ר ון-נוימן, לפיו כל תכנית מתבצעת בצורה רצופה, הוראה אחר הוראה ביע"מ אחד. בדור החמישי רוצים לבצע תכניות <u>במקביל</u>, כאשר הוראות שונות של התכנית מתבצעות באותו זמן, תחת פיקוח מרכזי של מנגנון בקרה משוכלל.

נמחיש את עקרון המקביליות בביצוע בעזרת חיבור של שני מספרים בעלי שש ספרות. בשיטה המסורתית, הרציפה, נהגנו לחבר את המספרים מימין לשמאל בעזרת מונה אחד:

```
 123456
+789123
 ------
 912579
```

בתחילה המונה מבצע 3+6, אח"כ 2+5 וכו'.

בחיבור מקבילי ניתן לחבר בבת-אחת את כל הספרות, באותו זמן, בתנאי שעומדים לרשותנו 6 מונים:

```
   1     2     3     4     5     6
 + 7   + 8   + 9   + 1   + 2   + 3
  --    --    --    --    --    --
   8     0     2     5     7     9
   0     1     1     0     0     0
```

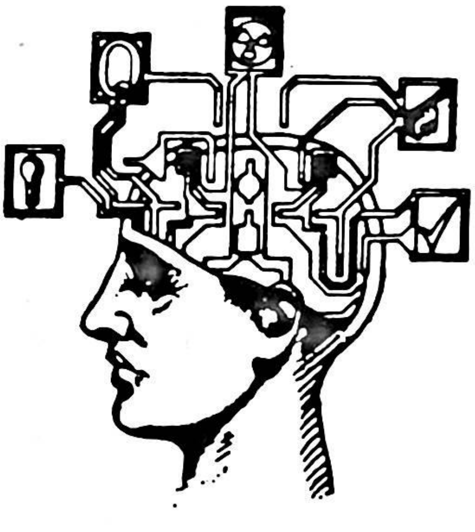
בשלב ראשון יחוברו כל זוגות המספרים, תרשם התוצאה ויצויין אם יש נגררת (CARRY). בשלב שני תחוברנה הספרות של הנגררות למונים שמשמאלן.

מחשבי הדור החמישי עשויים להשפיע בכיוון של שינוי התפישה לגבי תחום מעניין ביותר - <u>האינטליגנציה המלאכותית</u> (ARTIFICAL INTELLIGENCE). עד שנות השבעים ראו במחשב ובתכנה מכשיר חשיבה כללי, שעוזר לאדם בפתרון בעיותיו.. קיוו, שעם חלוף הזמן מערכת המחשב תשתכלל עד כדי כך, שיקום דור רובוטים דמויי אדם.

בשנות השמונים חל שינוי בתפישה, אשר מתבטא בחדירה לעומק בתחום צר יחסית, במקום להתפרס על שטחים רבים, ללא התמחות מסוימת. פותחו תכניות המבוססות על אגירת מידע רב מאד בתחום מקצועי ספציפי ופותחו תכניות לפי כללי חשיבה ייחודיים, אשר נקבעו בהתייעצות עם מומחים באותו תחום מקצועי.

ניקח לדוגמה את תחום רפואת העינים, שבה פותחו מערכות בהתייעצות עם רופאי עינים מומחים. התוצאה, <u>מערכת מומחה</u> (EXPERT SYSTEM) בשם CASNET, אשר מופעלת בהצלחה רבה ע"י רופאים כלליים בקנדה ובארה"ב. רופאים כלליים אלה עובדים במקומות יישוב נידחים ורחוקים והם חייבים לטפל גם במחלות עיניים, כי אין להם קשר ישיר אל רופאי עיניים, או מוסדות רפואה מרכזיים. תכניות אלה אינן באות במקום הרופא, אלא באות לקצר את תהליך הבדיקה ולהציע דרכים למציאת שיטת הריפוי.

כסיכום עלינו לציין, שאין לחשוש מפני מצב בו מחשבי הדור החמישי יהפכו לרובוטים דמויי אדם. ונסיים בדבריו של ד"ר בעל-שם: "המחשב של הדור החמישי יחקה את המדען בעבודתו, אך לא את המדען המשחק טניס לאחר שעות העבודה...".



יש רק דמיון אך לא שוויון

<u>פרק 1 - שאלות</u>

1. מהו עיבוד נתונים?

2. מהם ההבדלים בין נתונים למידע?

3. מהם שלושת השלבים העיקריים בתהליך עיבוד נתונים?

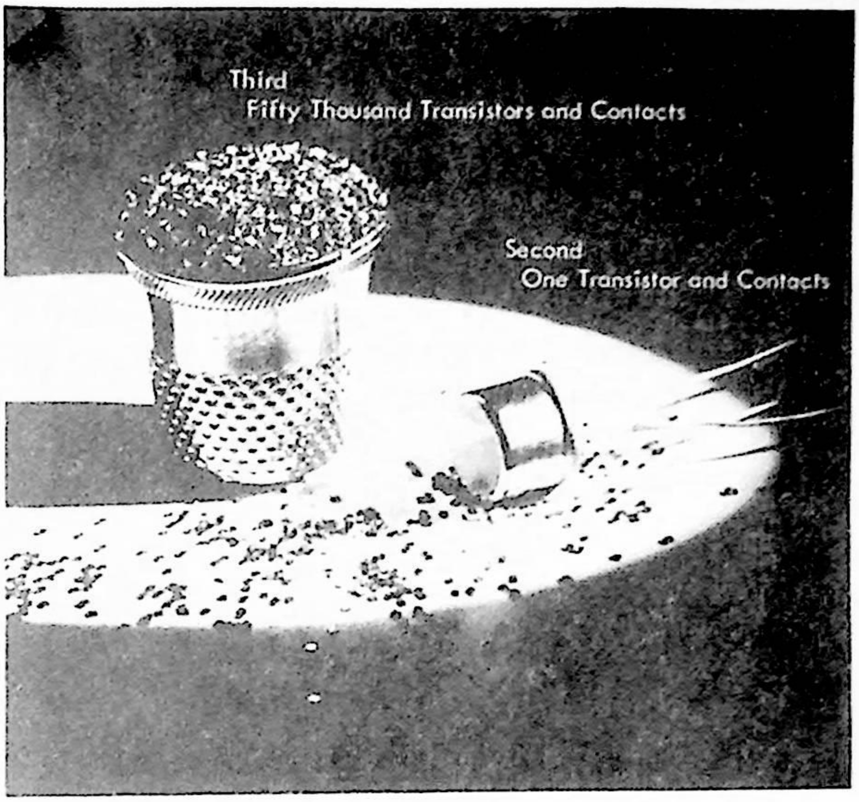
4. מהו שדה אלפא-נומרי? האם ניתן לבצע בו פעולות חשבוניות?

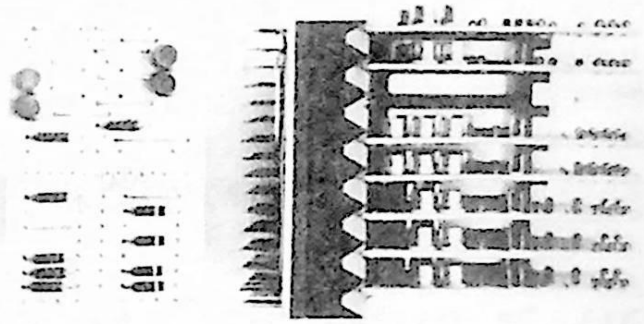
5. מהו המספר המירבי של השדות בתוך רשומה?

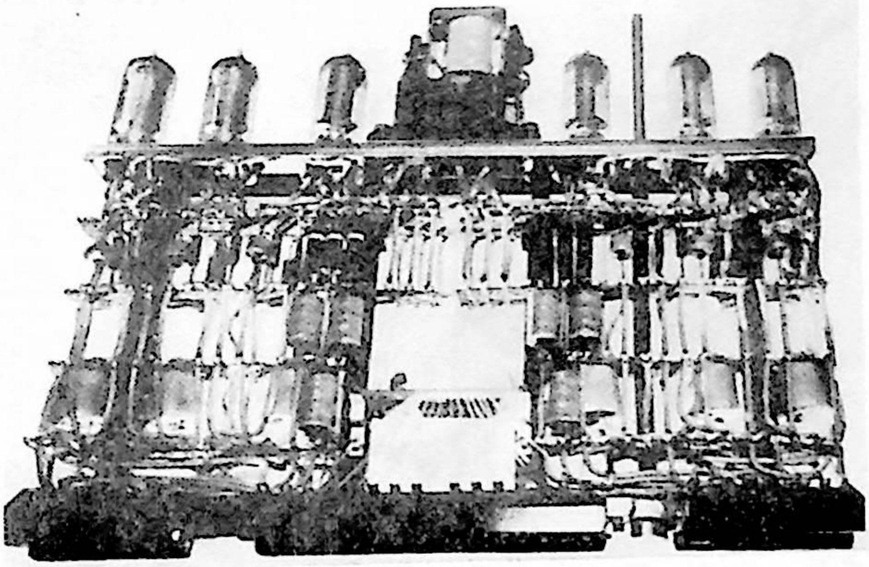
6. מה ההבדל בין הדור הראשון והדור השני של המחשבים?

7. הבא דוגמה לעיבוד נתונים <u>ללא</u> שימוש במחשב.

8. תאר בקצרה את תהליך העיבוד של ניהול חשבון הוצאות והכנסות שלך.







שלושה דורות של חומרה למחשבים
יחידות אלקטרוניות אשר ממלאות את אותה המשימה

פרק 2 – יישומי מחשב

אך לפני 10 שנים אפשר היה למנות בקלות יחסית את שימושי המחשב בתחומי התעשייה, המסחר, התחבורה, הצבא, המינהל הציבורי ועוד. נמצא בדרך כלל, שלמרות תחום הפעילות השונה, השימושים זהים כמעט, כמו למשל ניהול תקציב, ניהול מלאי, בקרת פרויקטים ועוד. היום התמונה שונה לגמרי! בימינו קשה למצוא תחום מסוים, שאליו טרם חדר המחשב. בשנים אלה ממש אנו חווים את אחת מהמהפכות החשובות בתולדות האנושות - את <u>מהפכת המחשב!</u> כבר לפני 10 שנים היו מדענים, שחזו את בוא המהפכה, אך אז היה המחשב עדיין בגדר דבר מה מופשט ורחוק, בלתי מוחשי ומסתורי כאחת.

קשה להאמין, אך השנה, בשנת 1984, יש יותר מחשבים בעולם מאשר בני-אדם! לפי הדיווחים האחרונים היו בעולם בשנת 1983 <u>כחמישה מיליארדים מחשבים</u> (מספר זה כולל גם את כל המיני ומיקרו מחשבים, המותקנים בכל מיני מכשירים מיוחדים, ומונים דיגיטליים, שחלקם משמשים לבקרה ולפיקוח אוטומטיים). המחשבים משרתים מפעלים, ארגונים, בנקים, רשויות מקומיות, משרדי ממשלה, חברות טלויזייה ותקשורת. על ידי השימוש בהם הולכים דפוסי חיינו ומשתנים כל העת.

לא ירחק היום בו סדר היום של משפחה טיפוסית ישראלית ייראה כך:

מר ישראלי מטלפן הביתה לאשתו ומודיע, שמחשב המפעל העביר את המשכורת למחשב הבנק. גברת ישראלי יוצאת לסופרמרקט כשהיא נושאת שני סלים וכרטיס זיהוי מגנטי בידה. בגמר הקניות היא מוסרת את הכרטיס לקופאית, אשר מעבירה אותו במכשיר מיוחד המחובר אל המחשב המרכזי של כל הבנקים. הסכום שמופיע ברישום הקופה, מנוכה אוטומטית מן החשבון של מר ישראלי ויתוסף לחשבון של הסופרמרקט. בדרך הביתה משלמת גברת ישראלי את חשבון הגז ומיסי העירייה בעזרת אותו כרטיס זיהוי מגנטי.

לאחר ארוחת הצהריים יושבים ילדי המשפחה לעשות שיעורי בית. גדי, הלומד בתיכון, זקוק למידע מסוים, שאיננו מופיע באנצקלופדייה. הוא מפעיל את מכשיר הטלויזייה, מחבר את המקלדת לשקע המיועד לכך בתחתית המקלט, ומקיש מספר אותיות. על המסך יופיע משפט, המאשר, שגדי מחובר למרכז המידע העירוני. גדי מקיש את שאלתו וכעבור מספר שניות מופיע על המסך מאמר הדן בנושא, שעליו שאל גדי.

מר ישראלי מגיע הביתה לקראת הערב ומזמין את המשפחה לצאת למסעדה. האם ללכת לאכול גלידה, בלינצים או אולי לבקר במסעדה סינית? הוחלט על מסעדה סינית, אך היכן יש בסביבה מסעדה סינית? מה התפריט ומהם המחירים? מר ישראלי מחבר שוב את המקלדת אל הטלויזייה ומתקשר אל מרכז המידע. בתוך שניות אחדות מופיעה רשימת כל המסעדות הסיניות בעיר עם ציון מספר טלפון, תפריט ומחירי המנות השונות. התשלום במסעדה יבוצע, כמובן, באמצעות הכרטיס המוכר. בערב, כשמגיעים הביתה, מחבר מר ישראלי שוב את המקלדת, והפעם עובר על הדואר האלקטרוני היומי, שהגיע אליו. כי יש לזכור, בעתיד לא ישלחו עוד מכתבים במעטפות, שבול מודבק על גביהן, אלא ישלחו ויקבלו תשדורות באמצעות המחשב של משרד התקשורת. כאשר אזרח מתחבר אל המחשב ומבקש את הדואר שהצטבר עבורו, נשלחים אליו כל המכתבים. המברקים וההודעות הישר אל מסך הטלויזייה, אך יתכן שצעירים עדיין יעדיפו לקבל דואר אישי מאוד במעטפה צבעונית ועל נייר מכתבים מבושם...

סדר יום כזה למשפחה כזאת לא נוכל למצוא עדיין בישראל וגם לא בארץ אחרת, אולם נעשו צעדים משמעותיים בארה"ב, בבריטניה ובצרפת לקראת עידן המחשבים. מומחים מעריכים, שעד סוף 1985 יעשו כל הסידורים המתאימים בארצות הללו כדי להכניס את האומה לעידן המחשבים. ומה קורה היום בישראל?

הזכרנו, שהמחשב חודר לתחומים חדשים ומגוונים. נסקור לדוגמא מספר יישומי מחשב חדשים או יישומים, שהורחבו ופותחו בעשור האחרון, ככל שאפשרה הטכנולוגייה.

2.1 המחשב ברפואה

1. המחשב לסיוע בניהול הרפואי

מחשבי קופת החולים הכללית חולשים היום על שמונה בתי-חולים, מעל ל-1000 מרפאות, על 30,000 רופאים, אחיות ועובדי מעבדה ומינהל ומעל שני מליונים של חברים.

כאשר מגיע חולה לחדר המיון ומחליטים לאשפזו, הוא נשלח למשרד הקבלה. במשרד הקבלה של בית החולים מוזנים למחשב הפרטים האישיים של החולה, שרופאי חדר המיון החליטו לאשפזו. פקיד הקבלה בודק באמצעות המחשב את התפוסה בכל מחלקה, ובהתאם לכך נשלח החולה לפנימית ג' או לכירורגיה א', הכל לפי העניין. מדפסת מדפיסה סדרה של תוויות ובהן הפרטים האישיים של החולה. תויות כאלה מודבקות על כרטיס החולה, על מבחנות המעבדה ועל גליון הרישום. ההדפסה אחידה ומסודרת ומונעת טעויות, שנגרמו בעבר בגלל כתיבה ידנית במחלקות השונות של בית-החולים. לאחר החלמתו עוזב החולה

את ביה"ח, אך פרטיו האישיים ופרטי הטיפול בו נשמרים במערכת המחשב, ואם הוא חוזר למרפאת-חוץ לביקורת, מקבל רופא המרפאה באמצעות המסוף את הגליון הרפואי של החולה, אשר כולל את פרטי המחלה ואת פרטי הטיפול שקיבל, בעת ששהה בבית-החולים, והוא יכול להוסיף פרטים בהתאם לעניין. גם במקרה של אשפוז בבית-חולים אחר מועבר המידע ישירות מן המחשב המרכזי של קופ"ח אל אותו בית-חולים.

הטיפול בתרופות נעשה גם כן על ידי המחשב. לטיפול זה שני היבטים:

- <u>אספקת תרופות</u> מרכזית לכל היחידות של קופ"ח. תכניות מחשב מיוחדות מנהלות את הבדיקה של מלאי התרופות, שומרות על רמה סבירה וכלכלית של אחסנה ודואגות להזמנה אוטומטית במקרה הצורך. הקצאת התרופות לבתי החולים ולמרפאות נעשית גם היא על ידי תכנית מחשב.

- <u>מינון</u> התרופה לחולים. בקרוב יוכל כל רופא לדאוג למינון נכון של התרופות בעזרת המחשב. המסוף אשר במחלקה יציג בחלק העליון של המסך את פרטי החולה ופרטי מחלתו, ובתחתית המסך תוצג רשימת תרופות עם מינון אפשרי. הרופא יוכל לקבוע, איזו תרופה תנתן לחולה, באיזו כמות ובאיזו תדירות. יופיעו גם שמות של תרופות היכולות לשמש תחליף לתרופה, שחסרה במקרה, או אולי שהחולה אלרגי אליה.

- <u>המחשב לעזרת הרופא</u>

במערכת הניהול הרפואי של קופ"ח הכללית פועלים מחשבים לשימוש כללי. בנוסף אליהם קיימים מחשבים רבים בעולם הרפואה לשימושים מיוחדים. דווקא בשטח זה צועדת בראש חברה ישראלית, שזכתה להכרה בינלאומית - חברת "אלסינט".

אחת מהמערכות המפורסמות של חברת אלסינט נקראת <u>טומוגרף</u>. זוהי מערכת המאפשרת לבצע סריקה ומיפוי מדוייקים של מוח האדם והצגת הממצאים על מסך או על מדפסת. גילוי גידולים במוח היה בעבר מבצע מסובך ומכאיב ולא תמיד הוכתר בהצלחה. בעזרת הטומוגרף ניתן למפות את המוח בתוך זמן קצר ובלא ניתוח בראש החולה. מערכת נוספת לבדיקת הרקמות הפנימיות, שיוצאת לשוק בימים אלה, היא מערכת "נמר". בבדיקת הרקמות של האברים הפנימיים השתמשו עד כה בקרני רנטגן, אשר חשפו את הנבדק לעיתים לקרינה חזקה או לקרינה בכמות רבה, אשר עלולה לגרום נזק לגוף. המכשיר החדש מאפשר את בדיקת

האברים הפנימיים ללא קרינה כלל אלא באמצעות מגנט רב עצמה. והיכן תרומתו של המחשב? - המחשב משמש בבקרת המערכות, ובניתוח נתוני המדידה ובהצגת התוצאות שנעשים באמצעותו.

2.2 תכנון ויצור בעזרת מחשב - תיב"ם

מומחים מעריכים, כי לקראת סוף 1985 יבוצעו 90% מעבודות השרטוט הטכני בעזרת תכנה הנקראת תיב"ם - **CAD** (COMPUTER AIDED DESIGN), ועד סוף העשור כ-70% של מפעלי התעשייה בעולם ישתמשו בטכניקה כלשהי של **CAM** (COMPUTER AIDED MANUFACTURING).

כיצד פועלת שיטת התיב"ם?

המערכת מבוססת על מצג גרפי מיוחד שמחוברים אליו עט חשמלי ומקלדת בעלת קלידים רבים. באמצעותם אפשר להפיק ציורים וגרפים שונים, תוך הזנת נתונים והוראות מתאימות (ציור 2-1). תווין, המחובר אל המערכת, יאפשר את קבלת הציור או השרטוט על נייר. למחשב קשורות יחידות קלט ופלט לאחסנת הנתונים, דיסקים וסרטים מגנטיים.



ציור 2-1 - מצג גרפי

תכנת תיב"ם כוללת שלוש קבוצות עיקריות של תכניות, בנוסף לתכניות השירות של המערכת:

1. <u>תכניות שרות</u>

- תכנית להבטחת קשר <u>הידברותי</u> (INTERACTIVE) בין האדם לבין המחשב.
- תכנית המטפלת בהקמה ובתחזוקה של מאגר נתונים.

2. <u>הכנת שרטוטים ומפרטים</u>

- תכניות ליצירת מודל גיאומטרי. המתכנן יכול לתאר את המודל בעזרת המסוף הגרפי, והמחשב מתרגם אותו למודל מתמטי ומאחסן אותו במאגר הנתונים לשימוש חוזר. האחסון נעשה ע"י שמירת המיקום של כל נקודה לפי הקואורדינטות שלה.
- תכניות להפעלת פונקציות קינטיות לצורך הצגה או סיבוב של המודלים סביב ציר מסוים.
- תכניות לשרטוט אוטומטי, אשר מאפשרות הפקת שרטוטים טכניים מפורטים לפי הנתונים אשר בתוך מאגר הנתונים.

3. <u>סיוע בתהליך הייצור</u>

- <u>בקרה ספרתית</u> (NUMERICAL CONTROL). תכניות אלו מפיקות תכניות עבודה למכונות כגון מחרטה, כרסומת ואחרות. הפלט הוא בסרט מגנטי או בסרט נייר מנוקב אשר מוזן למכונה.
- <u>תכנון תהליכים</u> (PROCESS PLANNING). תכניות אלו עוסקות בתכנון העבודה של מספר מכונות, המבצעות תהליך שלם מתחילת שלב ייצור מסוים ועד סופו.

נושא התיב"ם מיושם בישראל במספר ענפים, כשהבולטים בהם:

- תעשיית ייצור מטוסים - בתעשייה האוירית,
- תעשיית האלקטרוניקה - תדיראן
- תעשיית עיבוד המתכת - תע"ש,
- תעשיית הבנייה - חברת ACA.

שנת 1984 הוכרזה על ידי יצרני המחשב הגדולים כשנת מעבר לעידן המשרד האלקטרוני. במה דברים אמורים?

1. __ניהול המשרד__

במשרד טיפוסי אפשר להבחין בפעולות הבאות: פקידה כותבת מכתב, מכניסה אותו למעטפה, רושמת כתובת ושולחת אותו. העתק או שניים מהמכתב הנ"ל מתוייקים במקומות שונים. פקידה זו, או חברתה, עוסקת בהכנת רשימות של אנשים, שאליהם יש לשלוח חוזר מסויים, שנכתב זה עתה. אחרת מטפלת בשיחות הטלפון הנכנסות והיוצאות, ומישהי עוסקת בשכפול או בצילום חומר משרדי.

לאחרונה פותחו מערכות מחשב רבות או תכניות מחשב, אשר מכוונות למכן את כל הפעילויות, שתוארו לעיל. במשרד של שנת 1984 יעמוד מחשב קטן (ראה 2-2) ופקידה אחת תוכל לבצע בעזרתו את כל הפעילויות, שתוארו לעיל.



ציור 2-2 - המחשב למשרד

לאחר כתיבת המכתב במסוף תקבל הפקידה מזיכרון המחשב את שם הנמען וכתובתו, אשר יצורפו אוטומטית למכתב ויודפסו על גבי המעטפה. העתק המכתב יישמר בזיכרון המחשב ב"תיקים". אין צורך בצילום או בשכפול, כי כל מכתב ניתן לשליפה אל המצג של המחשב ולאחר בדיקה ניתן יהיה להדפיסו.

בזיכרון המחשב יאוחסנו דוגמאות של מכתבים טיפוסיים. אין צורך תמיד ליצור את המכתב בשלימותו מחדש, לפעמים ניתן

להרכיבו כך: ניקח את הרישא של מכתב מס' 20, את הגוף של מספר 33 ואת הסיפא של מכתב מס' 5 ונוסיף בין הקטעים או לסיום מספר משפטים, אשר מייחדים את המכתב החדש, כמו למשל נתונים מספריים וכמובן שם וכתובת הנמען.

תכונה חשובה נוספת: המחשב המשרדי יוכל לתקן שגיאות כתיב. כיום נמכרים מחשבים בהם יש מילון של 60,000 מלים, ואף יותר. במקרה של שגיאת כתיב מקרין המחשב את המלה השגויה על המסך בעצמת אור כפולה ומציע מספר מלים, בכתיב נכון, אשר אפשר להציבן במקום המלה השגויה. הפקידה תוכל להחליט אם "לשאול את הבוס" מה לכתוב...

עתה נותרה השאלה מדוע יש צורך להדפיס מכתבים? אפשר לשלוח ולקבל את המכתבים בשידור בין מחשבים, אשר קשורים זה לזה ברשת תקשורת ארצית.

המחשב יסייע בניהול שיחות הטלפון החל מרישום שיחות טלפון וחיוג אוטומטי וכלה בהשגת קשר עם המספר המבוקש, ואם יהיה צורך, יקליט את כל השיחה. המחשב יוכל לנהל יומני פגישות, לתת תזכורות ועוד. המחשב יהיה מסוגל להדפיס לפי הכתבה, ויוכל "לדבר" אלינו באוצר מלים של כמה מאות מלים. אך עם זאת, לא יוכל המחשב להיות יצור אנוש על כל המשתמע מכך.

2. המחשב ועורכי הדין

המבקר במשרד עורך-דין רואה לפניו ספרייה ענקית ובה מאות ואולי אלפי כרכים עבי-כרס, ועליהם רשומות המלים "פסקי-דין". פסקי הדין מסודרים לפי עניינים שונים ולפי סוג בית המשפט, שבו ניתנו אותם פסקי-דין. עורכי הדין חייבים להשתמש בספרים עבים אלה, כי בהופיעם בפני בית המשפט הם מסתמכים על תקדימים, שניתנו בפסק-דין זה או אחר.

בשנים האחרונות נעשה מאמץ לקבץ את כל פסקי הדין, אשר ניתנו לפני בתי הדין הרבניים. המיכון נעשה בעזרת המחשב של אוניברסיטת בר-אילן. היום ניתן לשלוף פסקי-דין בתוך דקות אחדות, לפי הנושא או התאריך שבו ניתנו, לפי בית-הדין שבו נפסקו וכו'.

2.4 סניף בנק ללא נייר

מספר בנקים בישראל מכינים חידושים מרחיקי לכת בעבור קהל לקוחותיהם: סניף בנק לשרות עצמי, ללא נייר, וללא פקיד קהל (כמעט).

בהכנסנו לסניף נפנה תחילה אל המסופים ולא אל הפקידים. לאחר הקשת המספר הסודי יאות המחשב לטפל בנו. נוכל לשאול, מה היתרה בחשבון עו"ש או במטבע חוץ, נוכל לבקש את רשימת ניירות הערך שלנו, את מחירי הניירות בבורסה, את ערך תיק ההשקעות שלנו וכו'. אולם לא די בכך, נוכל להעביר כספים מחשבון לחשבון, למכור ולקנות ניירות ערך, למשוך ולהפקיד כספים, בקיצור - כמעט הכל, אולי פרט לדבר אחד: לבקש הלוואה. לשם כך נצטרך להכנס אל מנהל הסניף, אשר כך מקווים אנו, יהיה לו פנאי להפגין כלפינו יחס אישי...

2.5 המחשב בשרות המשטרה

רב משרדי המשטרה בעולם מצויידים כבר במחשבים. חזיון נפוץ הוא לראות שוטר מדווח במכשיר קשר למוקד ומבקש לברר, האם רכב נטוש מספר 999-999 הינו גנוב או לא. במוקד ישנו מסוף, המחובר למחשב המרכזי של המשטרה, אשר מאפשר לברר מייד את כל פרטי הרכב. המחשב יכול להציג מספר מספרי רישוי של רכב, כאשר ניתנים פרטי זיהוי כמו צבע, מודל, חלק מהמספר ועוד. בירור פרטים יכול להעשות בשיטה דומה גם לגבי אנשים.

לפי תכנית "תירוש", שהוצעה לביצוע במשטרת ישראל, ייתכן שבקרוב יתממש התסריט הבא: המשטרה העירונית מפעילה שוטרי מקוף. במוקד ישנה מפה ועליה מסומנים גבולות כל האזורים ומכוניות המשטרה, המסיירות בעיר. קריאה מאחד האזורים יכולה להפנות את מכוניות השיטור לאזור הרצוי בתוך פרק זמן מינימלי. אם נתפס אדם, ניתן לברר במוקד, בתוך שניות, האם הוא מופיע ברשימת העבריינים, האם הוא מבוקש וכו'.

שימוש חדיש יחסית עושה המשטרה במחשב מיוחד, המותקן במסוקי המשטרה ביבשת אמריקה. במסוקים מורכב סוקר אופטי, המסוגל לגלות מטעים של צמחים נרקוטיים, מהם מייצרים סמים. הסורקים מופעלים ע"י המחשב כדי לשלוח קרניים בגלים קצרים אל הקרקע, ולקלוט את אורך הגל המוחזר. המחשב מנתח את הנתונים המתקבלים, ובהתאם לכך הוא קובע את מקום השדות החשודים. גילוי שדות כאלה בעזרת מסוק מאפשר את ריסוסם המיידי, לפני שהמגדלים מספיקים לקצור חלק מן היבול.

2.6 מחשב מדבר לעוורים

עד לשנת 1982 לא יכלו העוורים לנצל את שרותי המחשב לטובתם. פלט המחשב, אשר הודפס, או הוצג על מסכים, לא הגיע אל העוור.

בשנת 1982 נסתיים הפיתוח של מסוף מיוחד, שבו מותקן מנגנון, אשר יכול לדבר. בדרך זו אפשר להביא את תוצאות העיבוד של המחשב אל אדם, שאיננו יכול לראות. המסוף החדיש מאפשר לעוור להפעיל אותו תוך כדי השמעת טקסטים המופיעים על המסך באחת משלוש שפות: אנגלית, גרמנית או צרפתית. המסוף יכול לדבר בקצב של 300 מלים בדקה, אולם ניתן להתאים את קצב הדיבור לרצון השומע.
לנוחות המפעיל ניתן לקבל את הפלט בצורות שונות:

- הקראת הטכסט מהמסך בצורה שוטפת, עם סימני פיסוק או בלעדיהם,

- איות של מלים לפי בחירה,

- איות לפי אלפאבית פונטי כדי להקל את הזיהוי ע"י המאזין.

המסוף פותח ע"י יבמ, ומחירו גבוה במעט ממחיר מסוף רגיל. מסוף כזה נמצא כבר בארץ במרכז ההדגמות של חברת יבמ בתל-אביב.

2.7 פרות, עופות - והמחשב

מה הקשר בין פרות, עופות ומחשב מודרני? תשובה לשאלה זו ניתן לקבל במפעל "גרנות" ליד חדרה. מפעל "גרנות" משרת עשרות משקים בשומרון ובעמק חפר, הזקוקים לתערובת בעבור הרפתות והלולים שלהם.

מיכלים ענקיים של מרכיבי התערובת (סובין, סויה, תירס, קמח, חרבים וכו') תלוים בשורה בגובה רב. המיכלים מתחברים אל מכון ערבול ואל פתח יציאה, שתחתיו יכולה לעמוד משאית ענקית. המחשב זוכר עשרות תכניות שונות המכינות תערובת לפי דרישה - לפרה או לעוף, בהתחשב בגיל בעלי החיים, בעונת השנה ובנתונים אחרים המסופקים לו. הפעלת המכון הינה אוטומטית לגמרי - אדם אחד יושב מול לוח הפיקוד האלקטרוני, ובלחיצת כפתור הוא מכין תערובת מדוייקת, לפי התפריט המוזמן. בגמר תהליך ההכנה מוזרמת התערובת אל המשאית הממתינה מתחת לפתח היציאה של המיכל.

2.8 ה ר ו ב ו ט

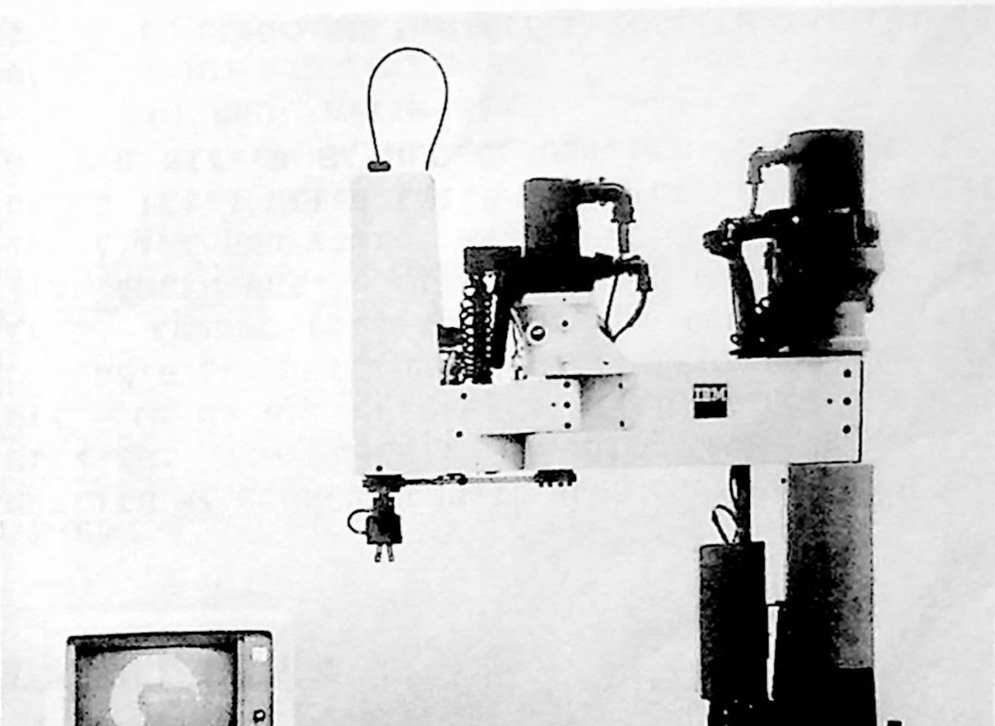
1. הרובוט בבית

לפני כשנה אחזה התרגשות גדולה בצוותי הקרקע של נמל התעופה בן-גוריון. גונבה לאזניהם השמועה שרובוט עומד

להגיע לארץ ושאנשי הטלויזייה יצלמו אותו. סמוך לבואו, הצטופפו מאות עובדי נתב"ג באולם הקבלה וראו את הרובוט מגיע...

הגובה - מטר וחצי, המשקל - 50 ק"ג. הרובוט אמר "הלו" ונע על גלגליו הקטנים ימינה ושמאלה, קדימה ואחורה. לאחר מכן נתנו לו מטאטא ויעה, והוא החל לנקות את אולם הנוסעים. ההצלחה היתה כבירה !

כיום מופעלים בעולם רבבות רובוטים, רובם בתעשייה ומיעוטם בבית. מה מסוגל רובוט לבצע בבית? ובכן, רב הרובוטים הביתיים מסוגלים כיום לטאטא, לשטוף רצפה, לשאוב אבק, לנקות שטיחים - ואפילו לשיר בתוך כדי העבודות המשעממות הללו. הרובוט מסוגל גם לשמור על הבית מפני פורצים וגנבים ומפני שריפה. הוא גם יכבה אורות או ידליק את המזגן בשעה המתוכננת וגם יפקח על התנור פן תשרף העוגה... כיצד הוא עושה זאת? בעזרת המחשב, כמובן. המחשב יכול להיות בנוי בתוך גוף הרובוט או להמצא במרחק מסויים ממנו ומשם לחלק את פקודותיו. ניתן לתכנת את המחשב כך שיורה לרובוט לשחק משחקים עם הילדים, או למשל, להעיר בכל בוקר את בני הבית במנגינה האהובה עליהם.



ציור 3-2

- רובוט

2. הרובוט בתעשייה

מדענים חוזים שימוש נרחב לרובוטים בתעשייה. בישראל נקלטו עד כה רק כמה מאות רובוטים, אך בעולם כולו מופעלים ועובדים רבבות רובוטים. "רובוט התעשייה" שונה במראהו מ"אחיו" הרובוט הביתי.

רוב הרובוטים התעשייתיים נראים כמלקחיים גדולים, אשר יכולים לנוע לכיוונים שונים, להרים חפצים ולהעבירם ממקום למקום או לסובב עצמים שונים סביב צירם.

ביפן פתחו בית-חרושת למכוניות, המתבסס כולו על עבודת רובוטים. באולמות חשוכים (לשם מה צריך הרובוט אור?) עובדים הרובוטים 24 שעות ביממה ורק בקצות האולם יושבים שני אנשים מול לוח פיקוד ענק ומשגיחים, "שהכל יהיה בסדר". העברת חלקי המכונית, צביעת החלקים, ההרכבה והחיזוק, הכל נתון בידי הרובוטים. תפוקת בית החרושת גדלה ב-300% לעומת תפוקת בית-חרושת רגיל בו עובדים בני אדם, וההוצאות התפעוליות נמוכות בהרבה.

3. הרובוט באזורי סכנה

רובוטים מופעלים בתעשייה במקומות אשר עלולים להוות סכנה לאדם, כמו למשל ליד תנורי היתוך. אך אפשר למצוא לרובוט תפקידים גם במקומות אחרים, שבהם האדם עלול להפגע. אחד מהשימושים האלה נעשה במשטרת ישראל, זהו הרובוט לסילוק חפצים חשודים. כשמתגלה חפץ חשוד, יוצאים חבלני המשטרה למקום עם הרובוט. הרובוט מופעל מרחוק, בעזרת גלי רדיו, לשם הזזת החפץ החשוד, ובכך הוא מאפשר לשוטרים המפעילים אותו להשאר מחוץ לאזור הסכנה.

שימוש ברובוטים מקובל גם בתחנות כח גרעיניות. חברת החשמל היפנית מפעילה כיום 7 תחנות כח כאלה. בשטח התחנות קיימים מספר אזורים בהם הכניסה אסורה לבני האדם בגלל עצמת הקרינה הגדולה, אשר היתה קוטלת כל אדם חי. ע"י הנחייה מרחוק מבצעים הרובוטים פעולות חיוניות, ובכך מבטיחים פעילות תקינה של התחנות.

<u>פרק 2 - שאלות</u>

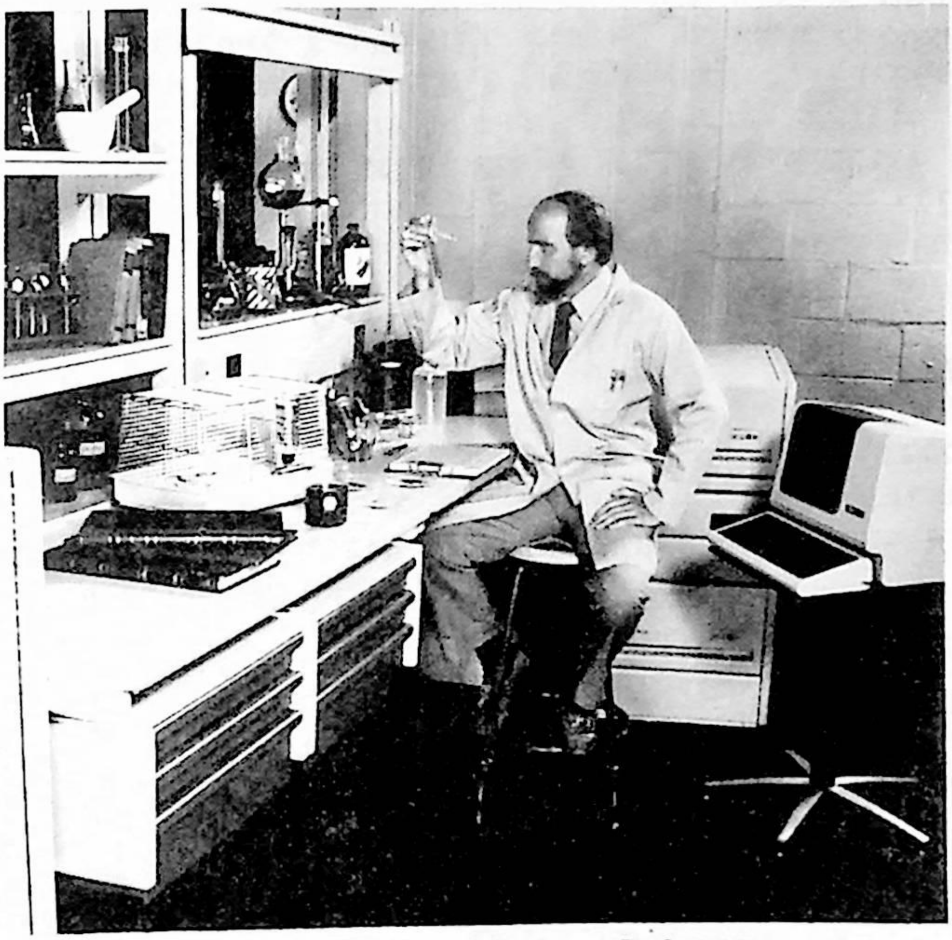
1. באילו דרכים מסייע המחשב לאדם בשטח הרפואה?

2. הסבר את המושג תיב"ם?

3. "בעוד חמש שנים לא נמצא פקידות במשרדים". הגב על ציטטה זו.

4. תאר בקצרה שימושים של הרובוט.

5. תאר והדגם כיצד השפיע המחשב על חיינו לפני חמש שנים, וכיצד הוא משפיע על חיינו כיום?

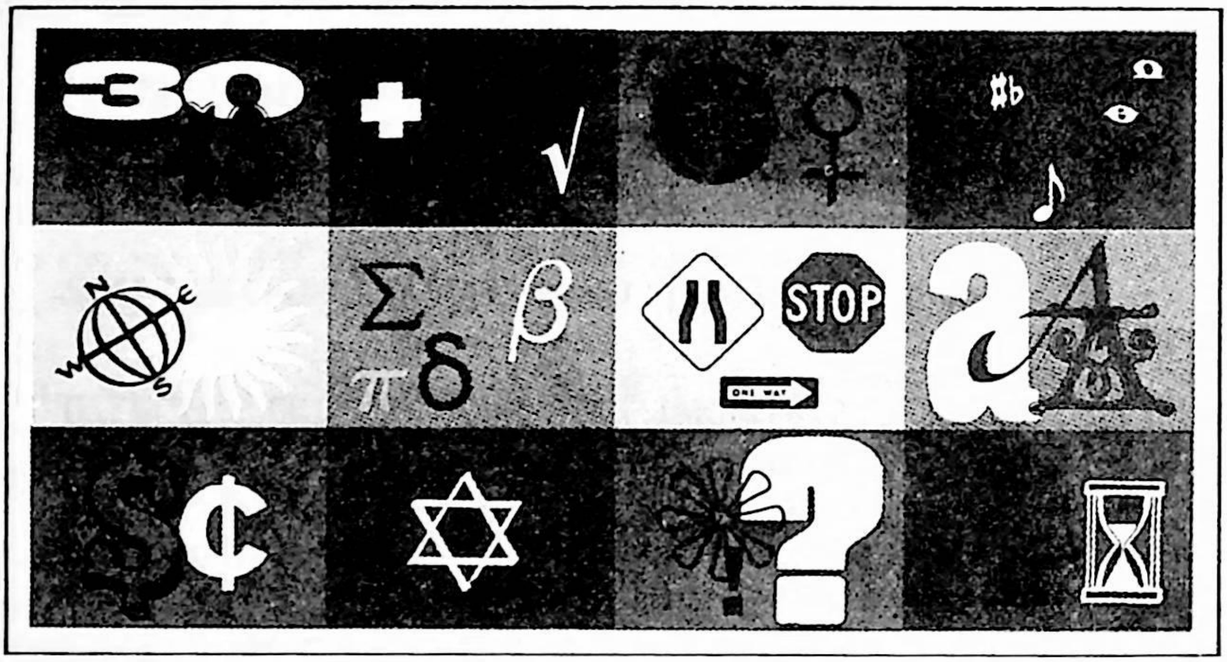


ציור 4-2 - המסוף (מחשב) במעבדה

פרק 3 - יחידת עיבוד מרכזית (יע"מ)

הבה נתבונן בציור 3-1, אשר מוצגים בו סימנים שונים. האם כולם מוכרים לך? ודאי יימצא ביניהם סימן אחד לפחות, שאינו אומר לך דבר. מהם הסימנים האלה? מה תפקידם?

כל סימן בציור הינו סמל. הסמלים נועדו להעביר מידע למתבונן בהם. אין זה אומר, שהסמל עצמו מהווה מידע כלשהו, אלא שהוא מייצג מידע ומציג אותו. כמובן קיים תהליך של למידה והכרה של סמלים, אך בזה לא נדון כאן.



ציור 3-1 - סמלים שונים

העברת מידע אל המחשב ועיבודו דומים במידה מסויימת לתהליך, שבו האדם מעביר מידע אל אדם אחר. נניח שיש לי חבר בצרפת, אשר איננו יודע עברית וברצונו להעביר אלי מידע כלשהו. הדרך המקובלת הינה כתיבת מכתב, במקרה זה בצרפתית, ושיגורו אלי. כאן נעשה שימוש במערכת סמלים הנקראת "האלף בית" של השפה הצרפתית, הכוללת את האותיות, הספרות וסימני הפיסוק השונים. בהגיע המכתב אלי אתרגם את המכתב לשפתי, לעברית, בעזרת מילון צרפתי-עברי או מכר היודע את שתי השפות.

בדרך דומה אפשר להעביר נתונים או מידע אל מערכת המחשב בעזרות שימוש במערכת סמלים מיוחדת. זו תהיה, כמובן, שונה ממערכת הסמלים המרכיבים את השפה העברית או מקבוצת האותיות הלטיניות. מהם אם כן, הסמלים "המובנים" למכונה מסוימת? דבר זה נקבע על ידי המתכנן והיצרן של אותה מכונה. למשל, המכשירים המיוחדים למשיכה אוטומטית של מזומנים (בנקט, כספומט וכו') "מכירים" כרטיסים פלסטיים מלבניים, שעליהם פס מגנטי, המכיל את המספר הסודי של בעל הכרטיס ופרטים נוספים אחרים. כאשר מכניסים למכונה את הכרטיס הפלסטי ומקישים את המספר הסודי, מתחיל המכשיר "לדבר" אלינו ומאפשר בתוך "שיחה" לבצע פעולות שונות כמשיכת מזומנים, בירור יתרה וכו'. מערכת הסמלים של כל מכונה הינה ה"שפה" שלה, בה אפשר לדבר אליה, ובה היא תדבר אלינו.

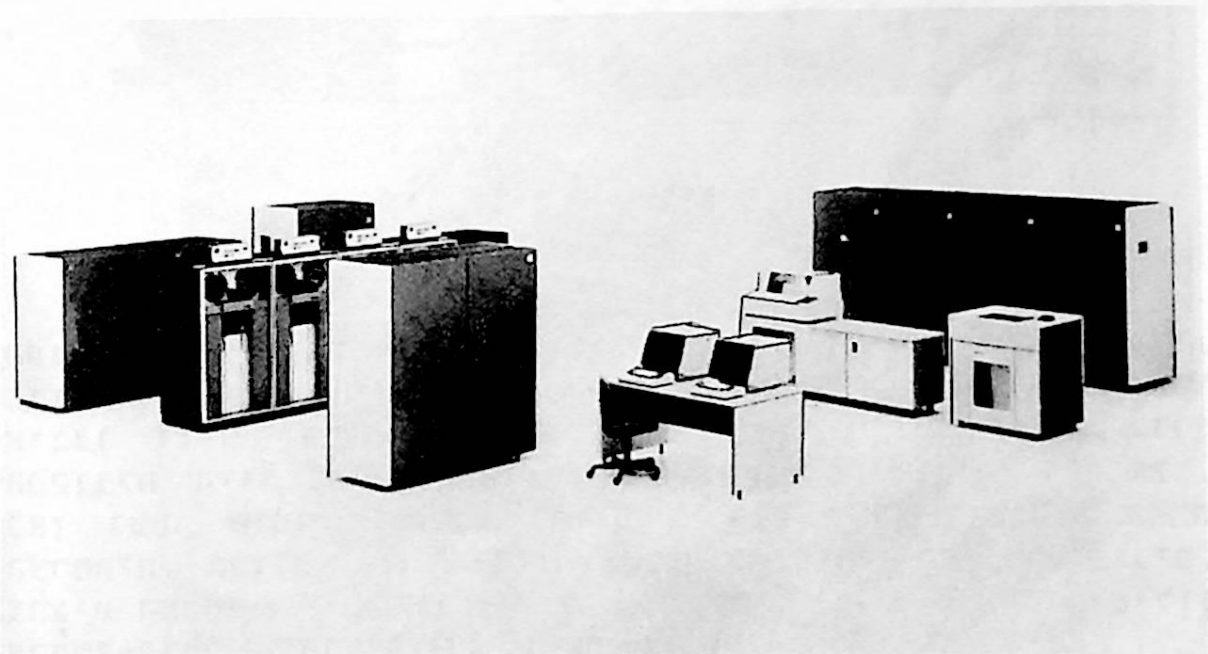
אולם, אל מי נדבר? האם לכל מערכת במחשב או לחלקים ממנו?

בפרק הראשון של ספר זה תוארו החלקים המרכיבים את המערכת. "לב" המערכת הזו הינה <u>יחידת העיבוד המרכזית</u> - <u>יע"מ</u> (CENTRAL PROCESSING UNIT - CPU), אשר מורכבת משלושה חלקים עיקריים:

1. <u>יחידת העיבוד</u> (PROCESSING UNIT), אשר מבצעת את החישובים השונים ואת הפעולות הלוגיות. היא נקראת לעתים גם <u>היחידה האריתמטית והלוגית</u> (ARITHMETIC AND LOGIC UNIT).

2. <u>יחידת הבקרה</u> (CONTROL UNIT).

3. <u>הזיכרון הפנימי</u> (INTERNAL MEMORY) או <u>אחסנה ראשית</u> (MAIN STORAGE). יש לציין, שלעיתים הזיכרון מופרד מהיע"מ ומהווה יחידה נפרדת.



ציור 2-3 - היע"מ ויחידות קלט/פלט

אין צורך לדעת הכל על החלקים השונים של היע"מ, אולם הסבר קצר על כל אחד מהמרכיבים יסייע לקורא להבין איך המחשב פועל. כך אנו מקווים, יסולק המסתורין, אשר אופף, שלא בצדק את המחשב וכל הקשור בו.

פרק זה יחולק למספר חלקים: בתחילה - הסבר קצר על השיטה הבינרית, אשר ישמש בסיס להבנת החלק הבא, אחריו - הזיכרון הפנימי, ולבסוף - המעגלים של יחידת החישוב והבקרה.

3.1 השיטה הבינרית

הבה נעבור עתה אל עולם המחשבים וננסה להבין מהי "השפה" בה משתמשים בעולם זה. נתבונן בנורת חשמל בחדר. נורה זו יכולה להמצא באחד משני מצבים: ברגע מסויים דולקת וברגע אחר - אינה דולקת. אומרים, שנורה זו פועלת בשיטה "בינרית" - שיטת מספרים בה משתמשים בשתי ספרות בלבד: 0 ו-1. נוכל לסמן מצב של נורה דולקת בספרה 1 ומצב בו הנורה אינה דולקת בספרה 0.

האלמנטים היסודיים של המחשבים כגון טרנזיסטורים, מעגלים חשמליים, ובעבר - טבעות ברזל, שניתנות למגנוט, פועלים גם כן בשיטה הבינרית. טרנזיסטור או מעגל חשמלי יכולים להמצא באחד משני המצבים 0 או 1, טבעת ברזל יכולה להיות ממוגנטת בכיוון אחד או בכיוון אחר, טרנזיסטור יעביר או לא יעביר זרם חשמלי.

כיוון שהמרכיבים של המחשבים פועלים לפי השיטה הבינרית, הבה נחזור ונזכיר את עיקרי השיטה הזאת (תיאור מפורט יותר מופיע בנספח לפרק זה). אולם לפני כן נרענן את זכרוננו בכמה פרטים, הקשורים בשיטה העשרונית.

בשיטה העשרונית משתמשים ב-10 סימנים 9,8,7,6,5,4,3,2,1,0. לכל אחד מהסימנים האלה, הנקראים ספרות, יש ערך אשר גבוה באחד מן הסימן הקודם לו. מספרים המופיעים בשיטה העשרונית הם למעשה צירוף של כמה ספרות, כאשר יש חשיבות רבה <u>למקום</u> של הספרה בתוך המספר. למשל, במספר 2872 הספרה 2 מופיעה פעמיים אך ערכה שונה בכל פעם. נבדוק זאת ביתר פירוט:

$$2 \quad 8 \quad 7 \quad 2$$

$$2000 + 800 + 70 + 2 = 2872$$

הספרה 2 מימין מייצגת שתי יחידות ואילו משמאל - שני אלפים!

בשיטה העשרונית כל מקום במספר יגדל לפי החזקות העולות של בסיס 10:

אלפים	מאות	עשרות	יחידות
10^3	10^2	10^1	10^0
1000	100	10	1

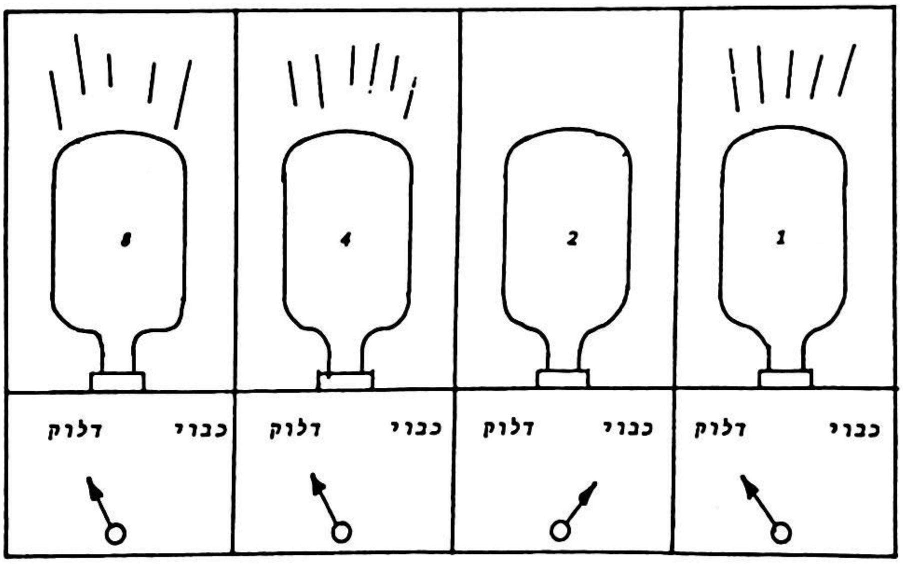
המספר 2872 ניתן לתיאור גם בדרך הבאה:

```
2       8       7       2
|       |       |       |
|       |       |       ---- 2x10^0 = 2x1    =    2
|       |       -----------  7x10^1 = 7x10   =   70
|       -------------------  8x10^2 = 8x100  =  800
---------------------------  2x10^3 = 2x1000 = 2000
                                               ----
                                               2872
```

בשיטה הבינרית כל <u>מקום</u> במספר גדל לפי החזקות העולות של בסיס 2:

"אלפים"	"מאות"	"עשרות"	"יחידות"
2^3	2^2	2^1	2^0
8	4	2	1

אפשר להציג זאת גם בעזרת 4 נורות חשמל, שניתן להדליק ולכבות. אנו נייחס להן את הערכים 1, 2, 4, 8 מימין לשמאל כפי שמודגם בציור 3-3.



ציור 3-3 - הצגה של המספר הבינרי 1101

בציור 3-3 יחסנו לכל נורה מספר אחר לפי החזקות העולות של 2. לכל נורה מתג, אשר יכול להמצא באחד משני מצבים: ON - דלוק, OFF - כבוי. נורות 1, 4, ו-8 דלוקות והן מייצגות את המספר 13 לפי השיטה הבינרית. במקום לציין מצבים של ON/OFF אפשר לציין מצב OFF ע"י הספרה 0 ומצב ON ע"י הספרה 1. לפיכך מצב הנורות בציור ניתן לתיאור ע"י הצירוף: 1101.

אפשר להציג את ערכו של המספר 1101 בדרך דומה להצגת המספר 2872.

```
1     1     0     1
|     |     |     |        o
|     |     |     ------ 1x2  = 1x1 = 1
|     |     |              1
|     |     ------------ 0x2  = 0x2 = 0
|     |                    2
|     ------------------ 1x2  = 1x4 = 4
|                          3
------------------------ 1x2  = 1x8 = 8
                                     ---
                                     13
```

בעזרת 4 הנורות אפשר להציג את המספרים בין 0 ל-15 כפי שמוצג בטבלה, בציור 3-4.

ייצוג עשרוני	ייצוג בינרי
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

ציור 3-4 - הצגת המספרים 0-15 בשיטה הבינרית

1024	512	256	128	64	32	16	8	4	2	1

ציור 3-5 - החזקות העולות של 2

בשיטה הבינרית הספרות הבינריות מסמנות, האם החזקה המתאימה של 2 קיימת או חסרה במספר רב-ספרתי. לפיכך כאשר רוצים לחשב את ערכו של מספר בינרי ולהעזר בטבלה, יש לסכם את הערכים העשרוניים של חזקות 2 המיוצגות ע"י הספרות השונות של המספר הבינרי. ניקח למשל את המספר הבינרי 1011011 ומעל ספרותיו נציין את הערכים העשרוניים המתאימים לפי טבלה 3-5. אח"כ נחבר את המספרים המופיעים מעל הספרות 1, ולא נתחשב במספרים שמעל האפסים:

64	32	16	8	4	2	1
1	0	1	1	0	1	1

ערך המספר: 91 (64 + 16 + 8 + 2 + 1 = 91)

כאן יש לציין נקודה חשובה: מתוך התבוננות בציור 3-4 מתברר, שאפשר לסמן את כל המספרים של השיטה העשרונית על ידי ארבע ספרות בינריות בלבד מ-0000 (0 לפי שיטה עשרונית) ועד 1001 (9 לפי השיטה העשרונית).

לפיכך, המספר 91 יכול להופיע בשתי צורות. צורה אחת: 1011011, בינרית,

או צורה שנייה, מקודדת:

9	1
1001	0001

השיטה בה מציגים ספרות דצימליות על ידי קבוצה בת 4 ספרות בינריות נקראת <u>סימון עשרוני בקוד בינרי</u> - BCD (BINARY CODED DECIMAL). למשל, המספר 1984 יופיע כך:

1	9	8	4
0001	1001	1000	0100

שיטת הצגה זו נקראת גם הצגת מספרים <u>בסימון דצימלי מקובץ</u> (PACKED-DECIMAL או בקיצור PACKED).

הביטוי האנגלי <u>BI</u>NARY DIGI<u>T</u>, שפרושו ספרה בינרית, הפך ל-<u>BIT</u> - מעין קיצור ..., וגם בעברית קוצר הביטוי <u>ספרה בינרית</u> ל-<u>"סיבית"</u>. מכיון שהמעגלים הזעירים של המחשבים נמצאים באחד משני המצבים, 0 או 1, נוהגים לקרוא להם גם BIT CELLS ובעברית, <u>סיביות</u> או <u>תגים</u>.

<u>3.2 הזיכרון הפנימי או אחסנה ראשית</u>

אנשים נוהגים לערוך רשימות באמצעים משונים ביותר. לעתים קרובות נראה אדם מוציא קופסת סיגריות מכיסו ומתחיל לרשום עליה בקדחתנות. לפעמים אפשר לראות בני-אדם היושבים במסעדה, ובתוך כדי ארוחה רושמים דברים שונים על מפיות הנייר או על דף התפריט.

כל "הרושמים" הללו נוהגים כך למעשה מאותה סיבה. הם חושבים, שלא יוכלו לזכור הכל "בראש" ולכן הם נעזרים בכל הבא ליד, בקופסת גפרורים, במפיות נייר וכו'. נוכל לומר, שלאדם שני סוגי זיכרון: זיכרון פנימי ("בראש") וזיכרון עזר חיצוני. הזיכרון החיצוני שלנו הוא לרב בכתב או בהקלטה: מחברת, ספר, פיסת נייר, מילון, פנקס למספרי טלפון, סרט הקלטה וכו'. אם מישהו אינו זוכר "על-פה" משהו, הריהו פונה אל רשימות, או אל ספרים ושם מוצא את מבוקשו.

<u>3.2.1 סוגי זיכרון</u>

גם כאשר מדובר במחשב - ניתן לחלק את סוגי הזיכרון לשתי קבוצות:

1. <u>זיכרון פנימי</u> (INTERNAL MEMORY) אשר נקרא גם <u>אחסנה ראשית</u> (MAIN STORAGE).

2. <u>זיכרון חיצוני</u> (EXTERNAL MEMORY) או <u>אחסנת עזר</u> (AUXILIARY STORAGE).

בפרק זה נתאר רק את הסוג הראשון של הזיכרון. בסוג השני נדון בפרק על יחידות קלט/פלט (סרטים ודיסקים מגנטיים) ושיטת האחסון בהם, ובפרק על ארגון נתונים.

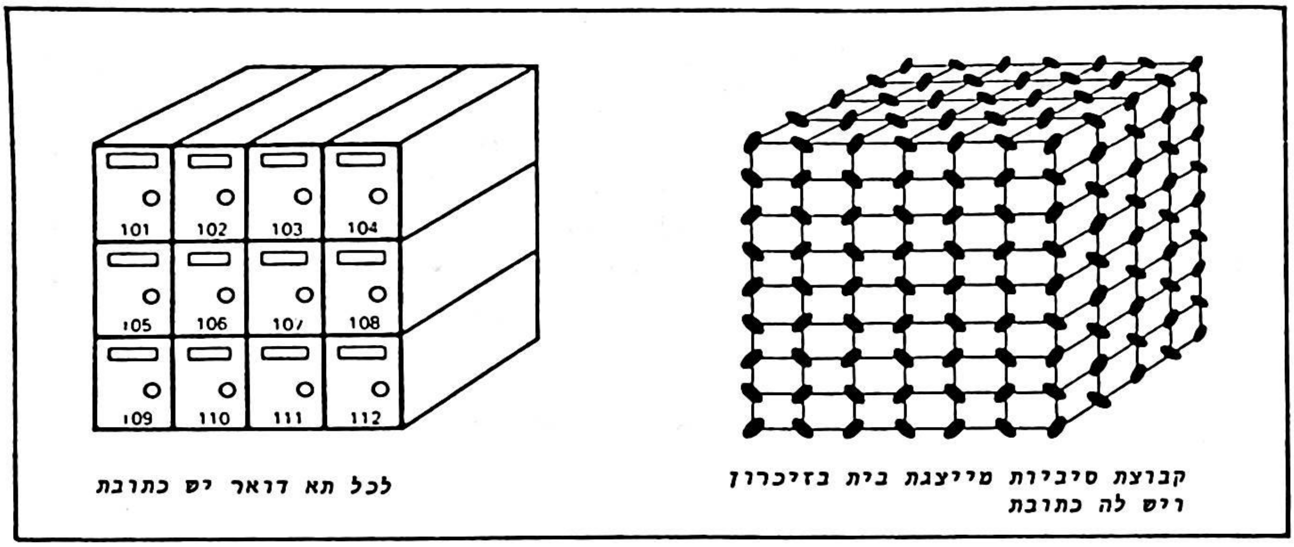
המרכיב העיקרי של האחסנה הראשית הלך והשתכלל במשך השנים. ממעגלים של שפופרות בסוף שנות ה-40, דרך טבעות ברזל פריט (FERRIT) נעירות הגיעו אל טכנולוגיות ייצור חדישות ביותר. עתה ניתן לבנות אלפי מעגלים אלקטרוניים על פיסה קטנה של סיליקון. מעגלים אלה הם המרכיבים היסודיים של הזיכרון הפנימי. כי זאת יש לזכור, אין אנו מאחסנים בזיכרון של המחשב משפטים, ספרות או סמלים, אלא מייצגים את התווים (אותיות, ספרות וכו'), שמהם מורכבים הנתונים.

יצרני המחשבים קבעו את הגודל האופטימלי של קבוצת סיביות, שבעזרתה ניתן לייצג תו אחד בזיכרון המחשב. חברת יבמ אשר משתמשת בקוד EBCDIC, (<u>סימול עשרוני בקוד בינרי מורחב</u> - EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE) קבעה יחידה בת 8 סיביות, הנקראת <u>בית</u> (BYTE) כיחידה יסודית של <u>זיכרון</u>. כל בית יכול להכיל תו אחד לפי טבלת EBCDIC. כשיטת הצגה מקוצרת של תוכן הזיכרון אמצה חברת יבמ את השימוש בשיטה <u>ההקסה-דצימלית</u> (HEXA-DECIMAL). יצרנים אחרים בחרו בקבוצה בת 6 סיביות כיחידה יסודית, שבעזרתה ניתן להציג תו אחד באחסנה. זוהי שיטת BCD (<u>סימול עשרוני בקוד בינרי</u> - BINARY CODED DECIMAL). שיטת ה"קצרנות" של מחשבים אלה היא השיטה <u>האוקטלית</u> (OCTAL). במחשבים שונים, ובעיקר במיקרו-מחשבים, מקובלת שיטת ASCII (AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE), בת 7 סיביות כדי להציג תו אחד באחסנה. על שיטות הספירה במחשבים, ראה גם בנספח א'.

כדי לציין את נפח זיכרון המחשב משתמשים במושג K (KILO), שפירושו בעולם המחשבים 1024 ($= 2^{10}$).

לפיכך:
1 KILOBIT = 1024 BITS
1 KILOBYTE = 1024 BYTES

כדי שאפשר יהיה לרשום נתונים ולאתר אותם לאחר מכן לשם שימוש, יש לדעת את מקומם או "כתובתם". אפשר לדמות את הזיכרון למערכת דמויית תיבות דואר בסניף הדואר, כאשר הבתים הם תיבות הנתונים.



ציור 3-6 - תיבות דואר

על כל תיבה רשום מספר ולפיו ניתן למצוא אותה בקלות. בדרך דומה כל בית נושא גם כן <u>כתובת</u> (ADDRESS). כל בית יאחסן נתון מסוים, אשר יכול להיות תו, אות, ספרה או סימן. במערכות מסויימות הכתובת מתייחסת לקבוצה גדולה יותר של סיביות, הנקראת <u>מלה</u> (WORD). מלה יכולה להכיל מספר בן כמה ספרות או שדה שלם של נתון אלפנומרי.

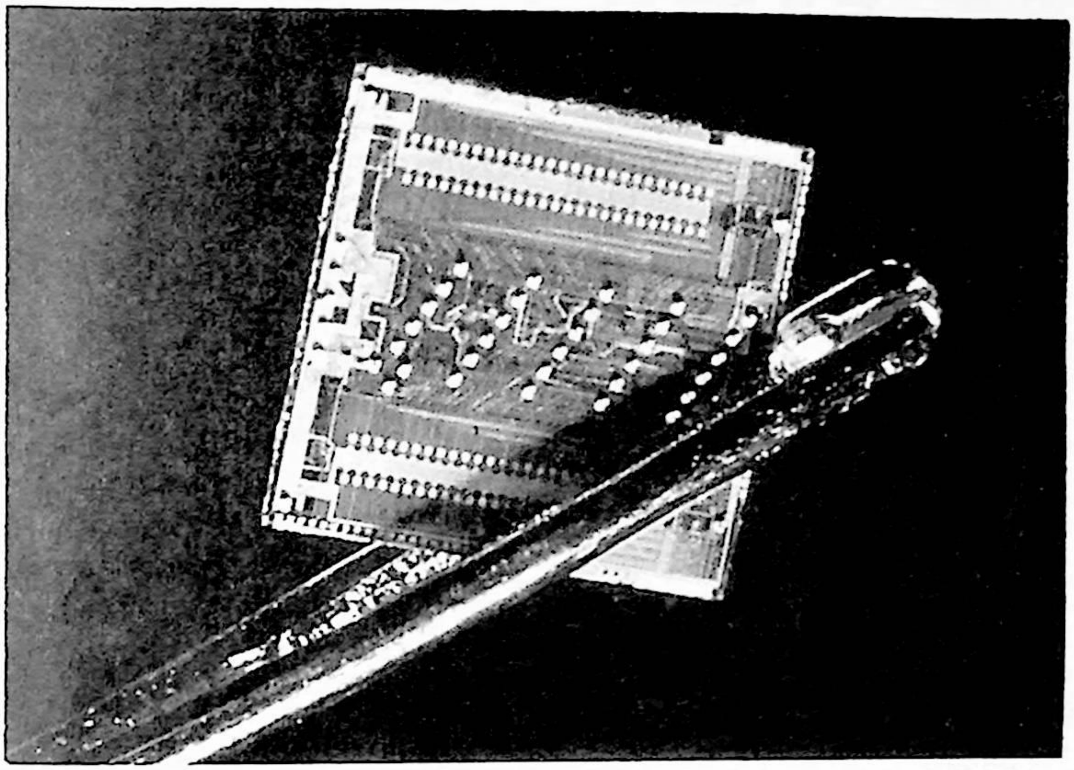
כדי להעביר נתון אל בית או אל מלה בזיכרון, חייבים לדעת את הכתובת שלהם. כתובת זו נקבעת בעת "עריכת" התכנית, כפי שנראה בהמשך. כאשר נתון "מועבר" אל כתובת מסויימת, התוכן שהיה בה קודם לכן נמחק. בזמן "קריאת" התוכן (הנתון) של כתובת מסויימת, הוא איננו נמחק, ואפשר לקרוא אותו שוב ושוב. ניתן לומר, שנתון המועבר בזיכרון איננו מועבר למעשה אלא <u>מועתק</u>, ועד שלא ייכתב נתון חדש באותה כתובת, הנתון הישן יישאר שם.

פרק הזמן הדרוש למחשב להעביר נתון אל כתובת מסוימת או ממנה נקרא <u>זמן גישה</u> (ACCESS TIME). פרק זמן זה הינו קצר מאוד ונמדד בננו-שניות (מיליארדית השנייה). כדי להעריך נכונה פרק זמן זה ניעזר בדוגמא הבאה: קרן אור עושה דרך של 300,000 ק"מ בשנייה אחת. במשך ננו-שנייה אחת מתקדמת קרן האור ב-30 ס"מ בלבד.

<u>3.2.2 סוגי מעגלים בזיכרון</u>

ייצור מעגלי הזיכרון עבר דרך ארוכה מאז הופיע המחשב הראשון, עם אלפי <u>שפופרות הריק</u> (VACUUM TUBES) וקילומטרים רבים של חוטים. אולם בכל התחנות של אותה דרך ארוכה חיפשו היצרנים את המרכיבים, אשר יכולים לייצג בקלות רבה מצבים של 0 ו-1, ולשנות מצבים אלו במהירות רבה. נקודה חשובה נוספת: חייבת להיות גישה אקראית וישירה אל קטע כלשהו בזיכרון. מכאן הביטוי <u>RAM</u> או RANDOM ACCESS MEMORY, שפירושו <u>זיכרון בעל גישה אקראית.</u>

ייצור מעגלי הזיכרון הינו מורכב ביותר ולא כאן המקום לתאר אותו. אולם נציין, שבעזרת טכנולוגייה חדישה ביותר מייצרים כיום את יחידות הזיכרון __בשבבים__ (CHIPS). השבב עשוי מסיליקון, ועל פיסה קטנה ממנו מודפסים מרכיבי המעגלים האלקטרוניים (נגדים, טרנזיסטורים וכו').ההדפסה נעשית בטכניקה מיוחדת, שעיקרה __המזעור__ (MINIATURIZATION) ובעזרתה ניתן להרכיב 64,000 סיביות על שבב בודד אחד, שגודלו אינו עולה על 0.5 סמ"ק! (ראה ציור 7-3). בשנת 1984 הצליחו לראשונה לייצר במעבדות שבבים עם 1,000,000 סיביות.



ציור 7-3 - שבב עם 64,000 סיביות

היות שכל המעגלים נמצאים על שבב אחד, נקראת שיטת הייצור הזו שיטה __מונוליתית__ (MONOLITHIC). זהו ביטוי ביוונית, שפירושו "אבן אחת". השם של מרכיבי המעגלים הינו __מוליכים למחצה__ (SEMI-CONDUCTORS). בגלל הנפח הקטן של יחידות הזיכרון, ניתן היום למקם את היחידה בתיבה אחת, יחד עם המעגלים של יחידת העיבוד המרכזית.

קיימים סוגים שונים של זיכרון מסוג RAM, אולם לכולם תכונה משותפת אחת - בעת הפסקת חשמל תוכן הזיכרון נמחק.

סוג נוסף של זיכרון, שאינו נפוץ עדיין, הוא __זיכרון בועות__ (BUBBLE MEMORY) המבוסס על מגנוט של אלמנטים שונים, ולכן __אינו__ נמחק עם הפסקת החשמל.

כאן המקום להסביר מושג נוסף: __זיכרון לקריאה בלבד__ (ROM - READ ONLY MEMORY). הכוונה לתאי זיכרון מיוחדים, בהם מטביעים או צורבים תכניות מחשב או נתונים, בבית-החרושת או

במעבדה. תאי זיכרון מסוג זה אינם ניתנים למחיקה ע"י המשתמש, והם מכילים בדרך-כלל פקודות או סדרת פקודות חשובה, אשר הכרחיות לפעולה התקינה של אותו מחשב ונדרשת על ידי רוב או כל המשתמשים.

3.3 יחידת עיבוד מרכזית - יע"מ (CPU)

היע"מ מפקחת על כל מערכת המחשב ומבצעת את כל הפעולות החישוביות והלוגיות. כפי שניתן לצפות, היא מורכבת מרבבות רכיבים זעירים בעלי תפקידים מיוחדים. כמובן אין צורך להכיר את כל הרכיבים הללו ולדעת מהו תפקידו המדוייק של כל אחד ואחד מהם. אולם מבט כללי יותר יאפשר לקורא להבין, כיצד עובד המחשב, ובכך לתרום לסילוק המסתורין האופף, אפילו היום, כל נושא אשר קשור בנושא "מחשב".

יחידת העיבוד המרכזית מחולקת לשני חלקים עיקריים: יחידת הבקרה והיחידה האריתמטית והלוגית.

3.3.1 יחידה אריתמטית ולוגית

__היחידה האריתמטית והלוגית__ (ARITHMETIC AND LOGIC UNIT) מורכבת ממעגלים אלקטרוניים הדרושים לביצוע כל הפעולות המוטלות על יחידה זו. פעולות אלו מחולקות לשני סוגים: הסוג הראשון נקרא __פעולה אריתמטית__ (חישובית), שמאפייניה הם: חישובים שונים (חיבור, חיסור וכו'), השוואה אלגבראית ובדיקת הסימן של המספר, עיגול מספרים במקרה הצורך וכו'. חישובים אלה מתבצעים על נתונים מספריים שונים, אשר מוצגים בזיכרון בצורה בינרית או בצורה דצימלית (ראה פרק זה, סעיף 3.1).

סוג שני של פעילות נקרא __פעולה לוגית.__ פעולה זו יכולה לחול הן על נתון מספרי (למשל לקבוע, האם ערכו של מספר חיובי, שלילי או אפס) והן על נתון אלפנומרי (למשל השוואת שתי מלים, או השוואת שני שמות, לצורך מיון אלפא-ביתי). באופן כללי ניתן לומר, שהפעולות הלוגיות מקנות למחשב את היכולת __לקבוע__, אם תנאי מסויים מתקיים, כדי לבצע סדרת הוראות מסויימת, או אם התנאי אינו מתקיים, כדי לבצע סדרת הוראות שונה.

ביחידה זו מתבצעות 4 פעולות החשבון: חיבור, חיסור, כפל וחילוק. כל בעית חישוב נפתרת בתכנית המוזנת למחשב תוך שימוש ב-4 פעולות אלו.

המחשב, עם כל עצמתו, מסוגל לבצע רק פעולות אריתמטיות פשוטות ולא מורכבות. גם פעולות הכפל והחילוק, אשר נעשות במערכת הפנימית של המחשב, מבוססות על על טכניקה של חיבור וחיסור.

פעולות העלאה בחזקה והוצאת שורש, חישוב אינטגרלים וכו" מתבצעים בתכנית ע"י פירוק הפעולה לקטעים בסיסיים שבהם מצטמצמים בביצוע 4 פעולות החשבון.

השוואת נתונים (הבדיקה הלוגית) מתבצעת אף היא ביחידה זו. השוואת הנתונים היא הצעד הראשון לקבלת החלטה; הצעד השני -- ניתוח ההשוואה ובחירת הצעד הבא לאור ניתוח זה.

ההשוואה מתבססת על קריטריון כמותי: אם נתון אחד שווה לשני, גדול או קטן ממנו. ניתן להשוות נתונים הנמצאים בזיכרון המחשב: כתוצאה מהזנה חיצונית או כתוצאת חישוב.

בעזרת הוראות מתאימות ניתן לעבור לביצוע קטע התכנית המתאים לאור תוצאות ההשוואה שבוצעה. פעולה זו היא שמקנה למחשב את יכולת ה"החלטה".

מבנה היחידה האריתמטית והלוגית

היחידה האריתמטית והלוגית מורכבת מאוגרים (REGISTERS). המבנה שלהם דומה למבנה של תאי זיכרון (מלים או תווים), אולם הם נפרדים ממנו ומבוקרים בנפרד ע"י מערכת הבקרה. כאשר האוגרים מופעלים לשם פעולות אריתמטיות הם נושאים כינויים ופועלים בשיטה מיוחדת כפי שתוסבר להלן.

במחשבים הפועלים בשיטה הבינרית נעשות כל הפעולות החשבוניות ורב הפעולות הלוגיות באמצעות האוגרים. במחשבים הפועלים בשיטה העשרונית נעשות הפעולות, לגבי התכנית לפחות, בתוך הזיכרון. האוגרים שבהם נעשית הפעולה הם אוגרים פנימיים של המערכת האריתמטית. פעולות ההשוואה יכולות להתבצע בין שני נתונים הנמצאים באוגרים, בזיכרון, או כאשר נתון אחד נמצא באוגר והשני בזיכרון.

במספר מחשבים משרתים האוגרים לפעולות חשבוניות גם את מערכת הפיקוד של פקודות וכתובות, בעוד שבמחשבים אחרים הם מהווים קבוצה נפרדת. האוגרים בשרות היחידה האריתמטית הם מיוחדים למטרות מוגדרות או יכולים להחליף תפקידים:

1. <u>צובר</u> (ACCUMULATOR)

   אוגר זה קולט את תוצאת הסיכום של שני מספרים ושומר אותה. הסיכום עצמו מבוצע <u>במסכם</u> (ADDER) פנימי במחשב.

2. <u>כופל מנה</u> (MULTIPLIER-QUOTIENT)

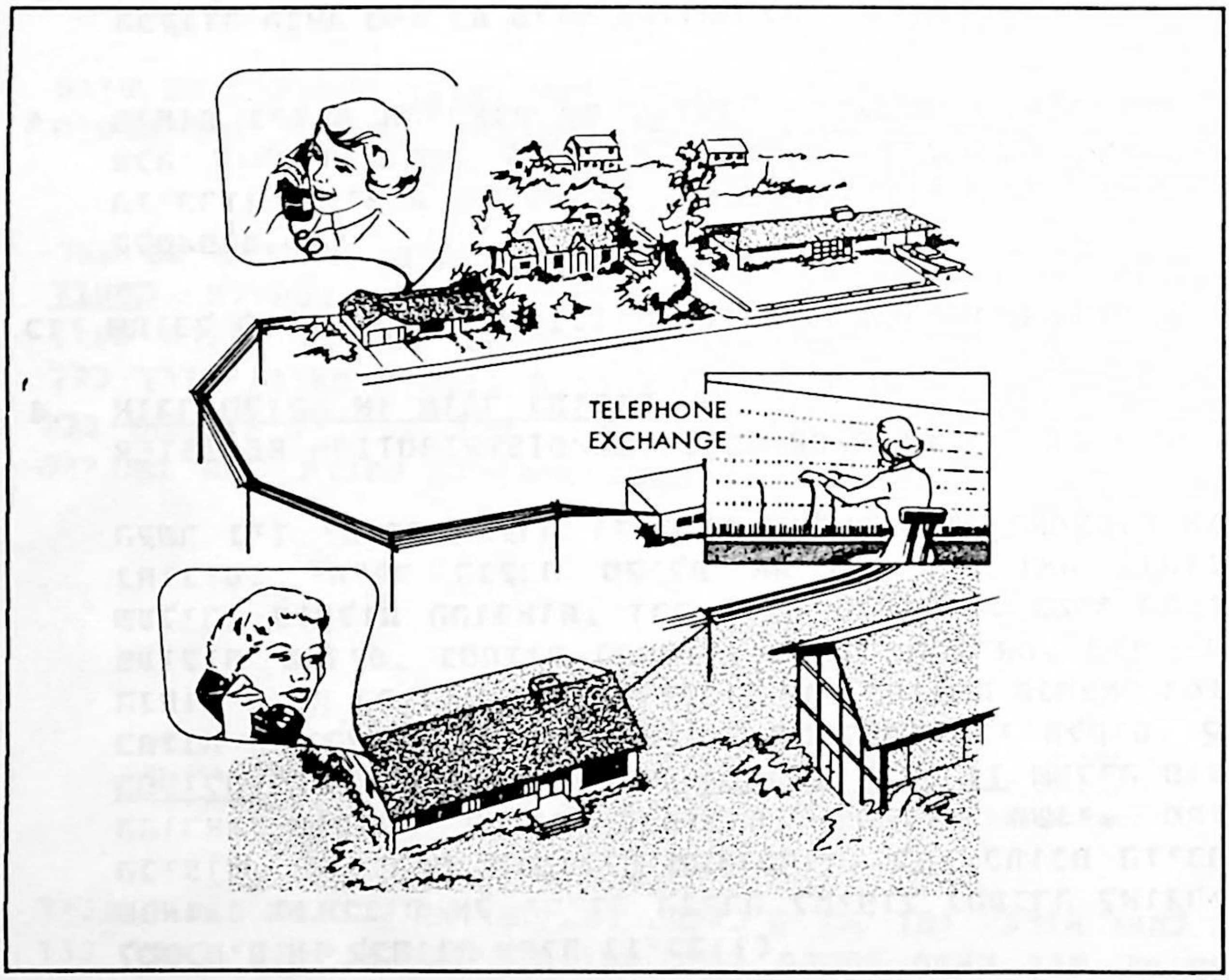
   תוצאת כפל של שני מספרים הנמצאים בשני אוגרים שונים תתן מספר אשר אורכו שווה לסכום ארכיהם, כלומר הוא יאחסן שני אוגרים רצופים.

בחילוק, נקבל מנה באורך של אוגר אחד אם נחלק נתון של שני אוגרים במחלק באורך של אוגר אחד.

לפי תאור זה אנו מנצלים זוגות של אוגרים לשם ביצוע פעולות כפל וחילוק. האוגר הימני המכיל את אחד מגורמי הכפל או את המנה בחילוק, הוא אוגר כופל מנה. האוגר השמאלי יכיל את הספרות הגבוהות של המספר במקרה של כפל, ובמקרה של חילוק את השארית.

3.3.2 יחידת הבקרה והפיקוח

<u>יחידת הבקרה</u> (CONTROL UNIT) דומה מבחינות רבות, למרכזיית טלפונים. המרכזייה מצויידת בכל הציוד הדרוש, כדי להעביר שיחות בין המנויים, המחוברים אליה. כך גם יחידת הבקרה, אשר מכילה את כל המנגנונים הדרושים להעביר נתונים מחלק אחד של מערכת המחשב אל חלק אחר בה (ראה ציור 3-8).



ציור 3-8 - מרכזיית טלפונים

מרכזיית הטלפונים מצויידת גם היא בכל הכלים, המפקחים על העברת פולסים מטלפון אחד לאחר: על צלצולים, על חיבור או על

ניתוק של מעגלים וכו'. יחידת הבקרה של היע"מ פועלת בשיטה דומה: היא מפקחת על יחידת הקלט והפלט, על העברת נתונים אל הזיכרון וממנה, ועל ביצוע פעולות חישוב והשוואה ועל פעולות אחרות.

יחידת הבקרה והפיקוח כוללת את המרכיבים המשמשים לניהול פעולות המכונה ומעקב אחרי ביצוען. תפקידיה העיקריים של יחידת הבקרה הם:

1. לפקח על ביצוע הפעולות האריתמטיות במערכת הפנימית של המחשב. פקודת החיבור למשל, היא רק פקודה אחת בתכנית, אבל ביצועה ביחידה האריתמטית דורש שורה ארוכה של צעדים בזה אחר זה.

2. לפקח על ביצוע הפקודות בסדר הנכון: "לדעת" מתי מסתיים הביצוע של פקודה מסוימת, לבקש מהזיכרון את הפקודה הבאה לפי התכנית ולהתחיל לבצעה.

3. לתאם את לוח הזמנים לביצוע הפקודות. יש להתחיל ביצוע הפקודה הבאה מיד עם סיום הקודמת לה.

4. תיאום ביצוע ההוראות עם פעולות קלט/פלט. הואיל ופעולות אלה דורשות גישה לזיכרון המחשב יש לפקח על הקצאת זמן הזיכרון לביצוע כל פעולה, כאשר העדיפות ניתנת לפעולות קלט/פלט.

כדי שתוכל למלא את יעודה, בנויה יחידה זו ממרכיבים אחדים:

1. <u>אוגר חלוקה או אוגר נתונים</u> -
STORAGE REGISTER/DISTRIBUTION REGISTER

הקשר בין יחידת הבקרה וזיכרון המחשב נעשה באמצעות אוגר נתונים. יחידת הבקרה מקבלת את ההוראות ואת הנתונים שעליהן פועלות ההוראות, דרך אוגר הנתונים, בשני מחזורי פעולות שונים. במחזור הראשון, מחזור ההוראה, קלט אוגר הנתונים את ההוראה מתוך הזיכרון לפי כתובתה הנמצאת במונה כתובת ההוראה. ההוראה השלימה מתחלקת לשני חלקים: <u>קוד הפעולה</u> מועבר לאוגר ההוראה <u>וכתובת הזיכרון</u> שעליה פועלת ההוראה מועברת לאוגר הכתובות. במחזור השני, מחזור הביצוע, מועברים הנתונים מהזיכרון, לפי כתובת הזיכרון שבאוגר הכתובות אל יחידת הבקרה לטיפול (העברה לאוגרים, למסכמים או לכתובת אחרת בזיכרון).

2. <u>אוגר כתובות</u> - ADDRESS REGISTER

יחידת הבקרה שולפת הוראות או נתונים מהזיכרון לפי הכתובת הנמצאת באוגר הכתובות. במחזור ההוראה מקבל אוגר הכתובות

את כתובת הזיכרון מתוך מונה כתובת ההוראה. במחזור הביצוע מקבל האוגר את הכתובת מתוך חלק הכתובת של ההוראה שבביצוע.

3. <u>מונה כתובת ההוראה</u> - INSTRUCTION ADDRESS COUNTER

במונה זה נשמרת הכתובת של ההוראה הבאה שיש לבצע. כאשר ההוראות מתקדמות לפי סדרן, המונה הזה מתקדם לפי מספר התוים או המלים הדרושים עבור כל הוראה והוראה. כאשר יש לבצע סיעוף (חריגה מהסדר הרצוף של ההוראות), מועברת למונה זה כתובת הוראה שאליה מכוון הסיעוף. לאחר מכן ממשיכה יחידת הבקרה לפעול בצורה הרגילה מאותה כתובת.

4. <u>אוגר הוראה</u> - INSTRUCTION REGISTER

באוגר זה מאוחסן קוד הפעולה של ההוראה שהמחשב מבצע. קוד הפעולה מועבר לאוגר זה במחזור ההוראה, מתוך אוגר הנתונים.

5. <u>מפענח</u> - DECODER

מטרתו לתרגם את קוד הפעולה, אשר באוגר ההוראות עם סיום מחזור ההוראה, ולהפעיל את המעגלים ביחידה האריתמטית והלוגית לשם ביצוע הפעולה הנדרשת.

<u>ביצוע הוראות ביע"מ</u> חייב להיות מתוכנן היטב ומתואם עם שאר חלקי המערכת, כדי להשיג יעילות מדבית בפעולת <u>המעבד</u> (PROCESSOR). פרקי הזמן במערכת המחשב נמדדים בעזרת שעון אלקטרוני פנימי, אשר פועל בתדירויות גבוהות מאוד, בדרך כלל עשרות מליוני פעימות בשנייה אחת. לכן נוהגים להשתמש בכל הקשור לזמנים "פנימיים" של מחשב ביחידות קטנות יותר משנייה אחת.

המונחים המקובלים הם:

0.1	שנייה -	1/10 שנייה,	או עשירית השנייה
0.001	שנייה -	1/1000 שנייה,	או אלפית שנייה (MILLI-SEC)
0.000001	שנייה -	1/1000000 שנייה,	או מיקרו-שנייה (MICRO-SEC)
0.0C0000001	שנייה -	1/1000000000 שנייה,	או ננו-שנייה (NANO-SEC)

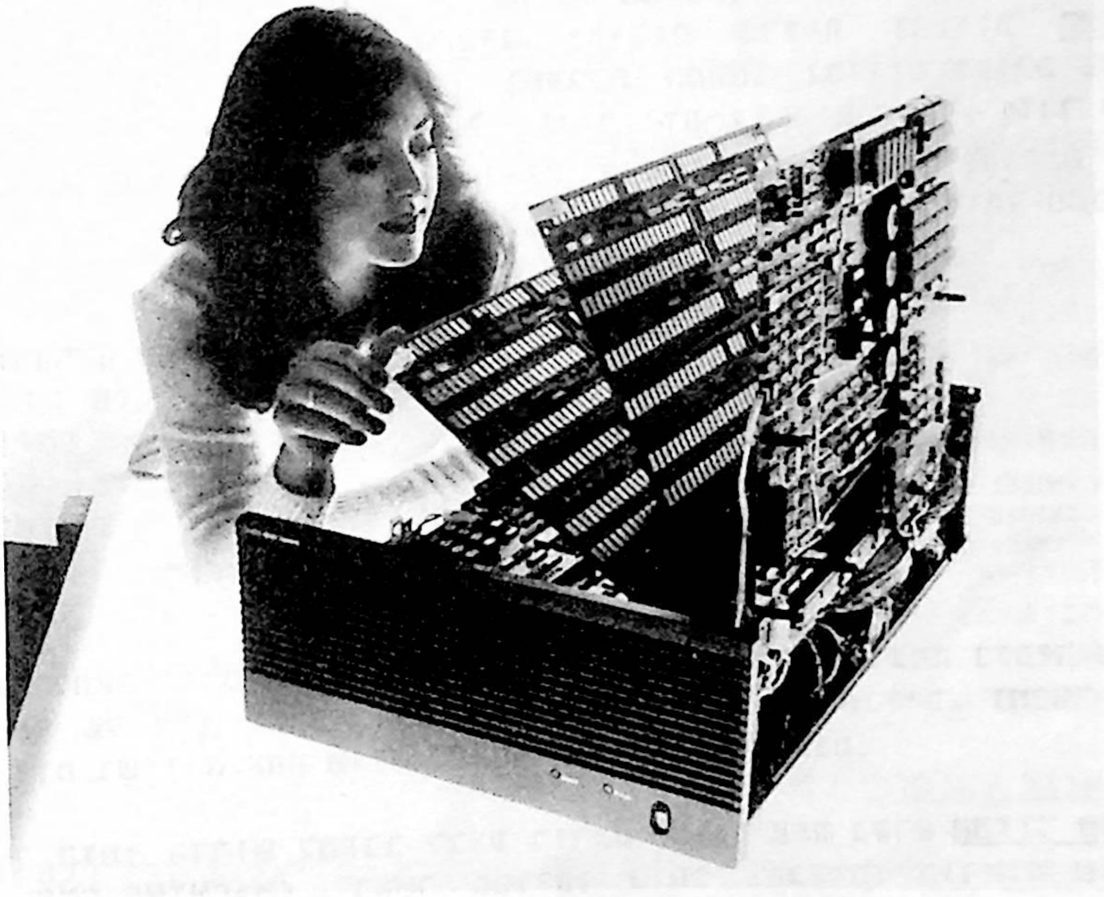
קשה לתאר אורכי זמן קצרים כאלה. לשם השוואה והדגמה נזכיר שניד עפעף של עין האדם נמשכת בד"כ 100 אלפיות השנייה, ומספר ננו שניות בשנייה אחת שווה למספר השניות ב-30 שנה.

משך הזמן הדרוש למעבד לבצע הוראת מכונה אחת נקרא <u>מחזור מכונה</u> (MACHINE CYCLE). כאשר תכניתן כותב בתכניתו הוראות שונות לביצוע, הן דורשות אורכי זמן שונים, כי אין דינה של הוראת

חיבור כדין הוראת חילוק למשל. כל הוראת תכנית כזו דורשת מספר שונה של הוראת מכונה.

אחד הכלים למדידת העצמה או כוח הביצוע של מעבד הוא משך הזמן הדרוש לביצוע הוראה אחת, למשל 2 מיקרו-שנייה. נוהגים לבטא זאת גם אחרת, מהו מספר הפעולות, שמעבד מסוגל לבצע בשנייה אחת: למשל, אם המעבד מבצע פעולה אחת ב-1 מיקרו-שנייה (מיליונית השנייה) אזי במשך שנייה אחת הוא יבצע מיליון פעולות. במקרה זה מדברים על יכולת ביצוע של <u>**1 MIPS**</u> (1 מיליון פעולות בשנייה - MILLION INSTRUCTIONS PER SECOND). במחשבים גדולים ומהירים נמצא ביצוע של 10 מליון פעולות בשנייה, ואף יותר.

מן הראוי לציין, שביצוע ההוראות במעבד אינו קובע בלבדית את תפוקת מערכת המחשב. רכיבי מערכת שונים, כמו ערוצים ויחידות קלט/פלט, מערכת הפעלה ותכניות המשתמש, שיטות עבודה ונוהלים, כל אלה תורמים לקבלת תפוקת המערכת. לפיכך אל לנו למדוד את עצמת המערכת (וזו טעות, שנעשית על-ידי רבים) על פי רכיב אחד ויחידי, אם כי חשוב. אם ננהג כך, נהיה דומים לאותה קבוצת עוורים, אשר בחנה פיל, וכל אחד תיאר והגדיר אותו לפי האיבר שמישש...



ציור 3-9 - "קרביים" של מיקרו-מחשב

פרק 3 - שאלות

1. מהם החלקים העיקריים של מערכת מחשב?

2. נתון המספר הבינרי 11100011. מהו ערכו לפי השיטה העשרונית?

3. הצג את ההפרש בין תאריך "המהפכה הצרפתית" לבין תאריך "גילוי אמריקה", לפי שיטת BCD.

4. מה ההבדל בין המושגים: BYTE, BIT.

5. הסבר את המושגים: כתובת, ROM, RAM.

6. פעולה לוגית - מהי?

7. האם קריאת נתונים מתוך הזיכרון מוחקת את הנתונים שנקראו? הסבר.

8. באילו יחידות מודדים את מהירות הביצוע של המעבד?

נספח לפרק 3 - שיטות ספירה במחשבים

כאשר מדברים על שיטות ספירה יש להבחין בין שלושה מושגים: הבסיס, הספרה הבודדת, וערך הספרה, שנקבע ע"י מיקומה בתוך המספר.

הבסיס קובע (או מגדיר) בכמה ספרות מותר להשתמש בשיטה מסוימת כלשהי. למשל, בשיטה העשרונית מותר להשתמש בעשרה סימנים: מ-0 ועד 9; בשיטה הבינרית הבסיס 2, מותר להשתמש בשני סימנים מ-0 עד 1; בשיטה האוקטלית הבסיס 8, מותר להשתמש בסימנים מ-0 עד 7; ובשיטה ההקסה-דצימלית הבסיס 16 ומשתמשים בסימנים מ-0 עד F.

הספרה הבודדת הינה חלק מסדרת סימנים, השייכת לאותה שיטה. בציור 3-3 מוצגות 16 שיטות עם הספרות שלהן. ערך הספרה בתוך מספר נקבע ע"י מקומה בתוך המספר. ערך זה מחושב ע"י הכפלת הספרה עצמה בחזקה המתאימה של הבסיס עצמו, בהתאם למקום הספרה במספר. ערך המספר עצמו הינו סכום כל המכפלות הנ"ל.

עליך לזכור, שכל מספר בחזקת 0 שווה 1 (פרט לספרה 0). למשל,

$7^0=1$, $83^0=1$, $1^0=1$. ונזכור :-

$a \neq 0$

- כל מספר מחולק בעצמו שווה 1 $\frac{a^n}{a^n}=1$

- מצד שני $\frac{a^m}{a^n}=a^{m-n}$

למשל $\frac{a^7}{a^3}=a^{7-3}=a^4$

אבל $\frac{a^5}{a^5}=a^{5-5}=a^0=1$

נציג למשל, את המספר 325 בשיטה העשרונית.

ערך הספרה	ערך מקומי	(10^0)	(10^1)	(10^2)	בסיס -->
		5	2	3	ספרה -->

$5\times10^0 = 5 \times 1 = 5$

$2\times10^1 = 2\times 10 = 20$

$3\times10^2 = 3\times100 = 300$

325 <---- ערך המספר כולו, לאחר חישוב.

מספר עם מעריך חזקה שלילי הינו שבר. למשל :

$10^{-1} = \frac{1}{10}$; $10^{-2} = \frac{1}{100}$ וכך הלאה.

וכך נציג את השבר העשרוני 92.81.

ערך מחושב	ערך מקומי	(10^{-2})	(10^{-1})	(10^0)	(10^1)	בסיס -->
		1	8	2	9	ספרה -->

$1 \times 10^{-2} = 1 \times \frac{1}{100} = 1.01$

$8 \times 10^{-1} = 8 \times \frac{1}{10} = 0.8$

$2 \times 10^{0} = 2 \times 1 = 2$

$9 \times 10^{1} = 9 \times 10 = 90$

92.81 <---- ערך המספר כולו, לאחר חישוב.

<u>השיטה הבינרית</u>

השיטה הבינרית תוארה כבר בפרק 3 של ספר זה ונחזור כאן רק על עיקרי הדברים:

ייצוג עשרוני	ייצוג בינרי
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111 וכך הלאה.

החזקות של מספר 2 (שהוא בסיס השיטה הבינרית) עוזרות להסב מספר בינרי לעשרוני.

נציג את החזקות של 2 ואת הערכים שלהם:

החזקה של 2 -	10	9	8	7	6	5	4	3	2	1	0
הערך של 2 בחזקה -	1024	512	256	128	64	32	16	8	4	2	1

הסבת מספר בינרי לשיטה עשרונית תיעשה ע"י רישום החזקות העולות של 2 מעל הספרות של המספר הבינרי. הכפלת המספר העשרוני במספר הבינרי שמתחתיו, וחיבור כל המכפלות תתן לנו את הערך של המספר.

למשל, נתון המספר הבינרי 1011, אשר נחשב את ערכו בשיטה העשרונית:

8	4	2	1
1	0	1	1

הפעולות מימין לשמאל:

(1x8) + (0x4) + (1x2) + (1x1)

11 שהם 8 + 0 + 2 + 1

כך אפשר לחשב בקלות רבה:

128	64	32	16	8	4	2	1
1	1	0	0	0	1	0	1

128 + 64 + 4 + 1 = 197

חיבור מספרים בינריים דומה לחיבור המספרים העשרוניים. כאן יש לזכור רק 3 כללים:

0 + 0 = 0
0 + 1 = 1
1 + 1 = 10 (אחד ועוד אחד הם אפס עם נגרר לשמאל)

נוכל עתה לחבר מספרים בינריים (ומקביליהם הדצימליים לביקורת):

```
  101      5      11      3
+ 110     +6    +100     +4
-----    ---    ----    ---
 1011     11     111      7
```

גם הכפל פשוט הרבה יותר בשיטה הבינרית מאשר בשיטה העשרונית. לוח הכפל של השיטה הבינרית נראה כך (זכור נא את לוח הכפל בן 100 המשבצות, אשר נדרש בשיטה העשרונית):

x	0	1
0	0	0
1	0	1

ע"י שימוש בחוקי החיבור ולוח הכפל, ניתן לבצע כפל של מספרים בינריים בקלות רבה. למשל,

```
  110       6
 x101     x 5
  ---     ---
  110
 000
110
-----
11110      30
```

הבדיקה:

	16	8	4	2	1
(30 = 16 + 8 + 4 + 2)	1	1	1	1	0

הסבת מספרים עשרוניים למספרים בינריים נעשית בדרך הפוכה מזו של הסבת מספר בינרי למספר עשרוני. בעוד שבהסבה של מספר בינרי

למספר עשרוני השתמשנו בפעולת כפל, הרי כאן נשתמש בפעולת חילוק.

שיטת ההסבה מתבססת על חלוקת המספר העשרוני בבסיס השיטה אליה רוצים להגיע (במקרה שלנו 2), רישום השאריות של פעולות החילוק והצגתן מימין לשמאל. כדי להקל על הרישום נסדר את הפעולות בטבלה בת שלוש עמודות: בעמודה השמאלית נרשום את המחלק, בעמודה האמצעית את המחולק (או את המנה), ובעמודה הימנית - את השארית. פעולת החילוק תמשך עד שנקבל בעמודה האמצעית מנה של אפס (נזכור שאנו נעבוד במספרים שלימים בלבד). לבסוף, נרשום את השאריות (שים לב, אם מחלקים ב-2 השארית הגדולה ביותר תהיה 1) בצורה אופקית ונקבל מספר בינרי. <u>יש להקפיד</u> שהשארית הראשונה (הספרה העליונה בעמודה הימנית) תהיה הספרה הימנית ביותר של המספר הבינרי.

נמחיש את כללי ההסבה בדוגמה. נמצא את ערכו הבינרי של המספר העשרוני 25:

שארית	מחולק (מנה)	מחלק (בסיס)
	25	2
1	12	2
0	6	2
0	3	2
1	1	2
1	0	2

1 1 0 0 1

25 : 2 = 12 (1)
12 : 2 = 6 (0)
6 : 2 = 3 (0)
3 : 2 = 1 (1)
1 : 2 = 0 (1)

בדיקה:

16	8	4	2	1
1	1	0	0	1

16 + 8 + 1 = 25

מספרים בינריים אינם נוחים לשימוש כי מספר הספרות בהם הוא רב. למשל, המספר 600 ייכתב ע"י עשר ספרות: 1001011000.

בגלל קושי זה נוהגים אנשי המחשבים להשתמש בשיטות כתיבה מקוצרות. נפוצות שתי שיטות, השיטה האוקטלית והשיטה ההקסה-דצימלית.

השיטה האוקטלית

בשיטה האוקטלית 3 ספרות בינריות מיוצגות ע"י ספרה אוקטלית אחת.

אוקטלי	בינרי
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

מספרים אוקטליים, גדולים מ-7 נכתבים כך :

עשרוני	אוקטלי	
8	10	
9	11	
10	12	
11	13	
12	14	
13	15	
14	16	
15	17	
18	20	
19	21	וכך הלאה.

הסבת מספר אוקטלי למספר עשרוני נעשית באופן דומה להסבה של מספר בינרי למספר עשרוני, רק שכאן עלינו לזכור מהן "העשרות", "המאות", ו"האלפים" של האוקטלי, שהן למעשה החזקות העולות של 8.

עשרוני	1000	100	10	1
אוקטלי	512	64	8	1
	8^3	8^2	8^1	8^0

למשל, המספר האוקטלי 125 יוסב למספר עשרוני כך :

(1 x 64) + (2 x 8) + (5 x 1) שהם:

85 = 64 + 16 + 5

הסבת מספר עשרוני לאוקטלי מתבצעת בשיטה הדומה לשיטת ההסבה הקודמת.

למשל, הסבת המספר העשרוני 600 לשיטה האוקטלית תיעשה בדרך הבאה:

שארית	מחולק (מנה)	מחלק (בסיס)
0	600	8
3	75	8
1	9	8
1	1	8
	0	

1 1 3 0

בדיקה:				
	0	3	1	1
	1	8	64	512

$$600 = (1 \times 512 + (1 \times 64) + (3 \times 8)$$

השיטה ההקסה-דצימלית

השימוש בשיטה ההקסה-דצימלית מאפשר לכתוב ספרה הקסה-דצימלית אחת במקום ארבע ספרות בינריות. נראה זאת בטבלה הבאה:

עשרוני	הקסה-דצימלי	בינרי
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
17	11	10001
18	12	10010
19	13	10011

וכך הלאה.

החזקות העולות בשיטה זו:

65536	256	16	1
16^3	16^2	16^1	16^0

המשתמשים בשיטה ההקסה-דצימלית נעזרים בדרך כלל בטבלאות. כאן נביא שתי דוגמאות להצגת הדרך של הסבה אל ומהשיטה העשרונית.

המספר A B 1 יוסב למספר העשרוני כך:

1	B	A
256	16	1

(A x 1) + (B x 16) + (1 x 256) שהם :

$442 = (1 \times 256) + (11 \times 16) + (10 \times 1)$

דוגמה להסבת מספר עשרוני להקסה-דצימלי:

מחלק (בסיס)	מחולק (מנה)	שארית
16	333	13
16	20	4
16	1	1
	0	

1 4 (13)
D

נקבל את המספר D 4 1 שערכו בשיטה הדצימלית 333. נוכל לרשום זאת כך:

$$14D_{16} = 333_{10}$$

לבסוף נציג את השימוש בשיטה האוקטלית ובשיטה ההקסה-דצימלית כשיטה של "קצרנות".

כתיבת מספר בינרי בשיטה אוקטלית נעשית על ידי חלוקת המספר הבינרי לקבוצות של 3 ספרות, החל מן הנקודה העשרונית.

למשל:

בינרי	111	010	100	001	011	1
אוקטלי	7	2	4	1	3	1

כתיבת אותו מספר בשיטה ההקסה-דצימלית תיעשה לאחר חלוקה לקבוצות של ארבע ספרות:

בינרי	0111	0001	0011	1011
הקסה-דצימלי	7	1	3	B

החסכון בכתיבה נראה בבירור. במקום מספר בינרי בן 16 ספרות, מקבלים מספר הקסה-דצימלי בן 4 ספרות. בגלל סיבה זו נוהגים יצרני המחשבים להשתמש בתכנית, אשר מדפיסה את תוכן תאי הזכרון בשיטה האוקטלית או ההקסה-דצימלית.



האם זהו כל הסוד!

פרק 4 - יחידות היקפיות למחשב

למערכת מחשב ניתן לחבר <u>יחידות קלט ופלט</u> (INPUT-OUTPUT UNITS) <u>ויחידות אחסון נתונים</u> (DATA STORAGE UNITS) שונות. מכנים אותן גם <u>יחידות היקפיות</u> (PERIPHERAL UNITS), כיון שהן "מקיפות" את יחידת העיבוד המרכזית, מזינות אותה וקולטות ממנה נתונים ומידע. מקובל גם לכנות את <u>היחידות</u> (UNITS) בשם <u>התקנים</u> (DEVICES), מלשון מיתקן, ולדוגמה "התקני קלט ופלט".

הבעיה, אשר עמדה בפני המתכננים של יחידות הקלט/הפלט היתה: כיצד להציג למחשב, אשר "מבין את שפת החשמל" (כי הריהו פועל עם מעגלים אלקטרוניים) את הנתונים, שהאדם מבקש לעבד בעזרתו. הנתונים הללו מוצגים לאדם בעזרת הסמלים הידועים של סימני כתיבה: אותיות, ספרות, סימני פיסוק וכו'. צד נוסף של הבעיה קשור בעובדה, שהמחשב כבר "הבין" את הנתונים של האדם ועיבד אותם בעזרת המעגלים האלקטרוניים שלו. כיצד להציג את תוצאות העיבוד, שנמצאות בזיכרון של המחשב, לאדם, אשר אינו מבין את "שפת החשמל"?

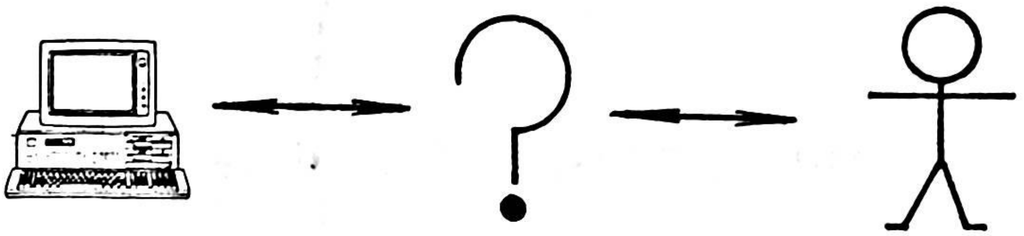
הבה נעצור רגע, ונברר מהי בעצם שפת החשמל. בצורה פשוטה נוכל לומר, ששפת החשמל ניתנת להצגה על ידי שני סימנים בלבד: סימן אחד יסמל שזורם זרם במעגל חשמלי מסוים (נניח הסימן '1'), וסימן שני יסמל, שהזרם אינו זורם באותו מעגל (נניח הסימן '0').

הבעיה "ההידברותית" בין אדם למחשב והפתרון שלה, מוצגים בציור 4-1.

הפתרון הרצוי הוא אמצעי או התקן, אשר, מצד אחד, יכול לתרגם <u>תמליל</u> (TEXT), המורכב מסימני כתיבה המובנים לאדם לסימנים של 1,0 (מצבים של זרימה או ניתוק של זרם חשמלי) במקרה של פעולת קלט. מצד אחר - בעת פעולת פלט; ההתקן הזה יכול לתרגם מצבים של מעגלים חשמליים (המצויינים על ידי 'אחדי-ים ו-'אפסי-ים) לסימנים, אשר ירכיבו תמלילים המובנים לבני האדם.

רוב יחידות הקלט והפלט הן התקנים כאלה, אשר מבצעים את "התרגומים" הנדרשים בצורות שונות ובאופנים שונים. בפרק זה נעמוד על תיאור יחידות הקלט והפלט השכיחות ונסביר את אופן פעולתם.

ה ב ע י ה



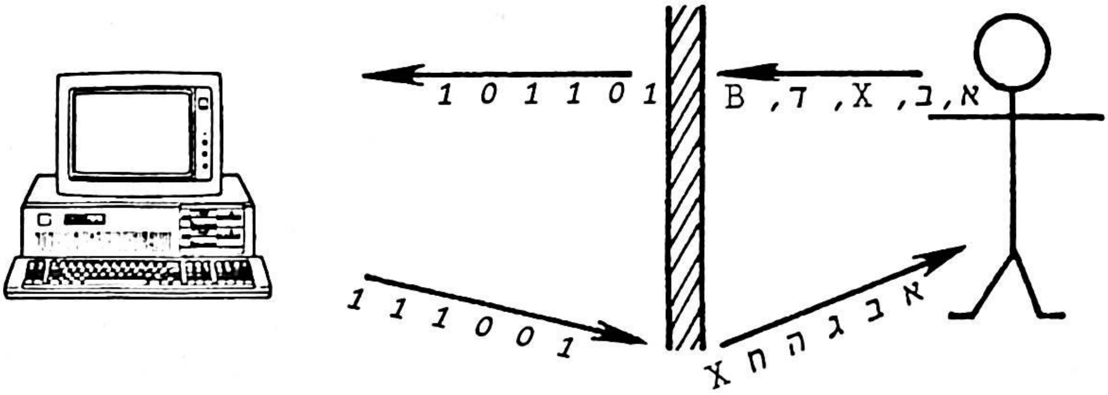
אדם

מבין שפה, שסימניה
א,ב,ג, A,B,C
1,2,3,?,: וכו'

מחשב

מבין שפת החשמל, שסימניה
'ו' (יש זרם חשמלי במעגל)
'ס' (אין זרם חשמלי במעגל)

הפתרון הרצוי



ציור 4-1-"הידברות" בין אדם לבין מחשב

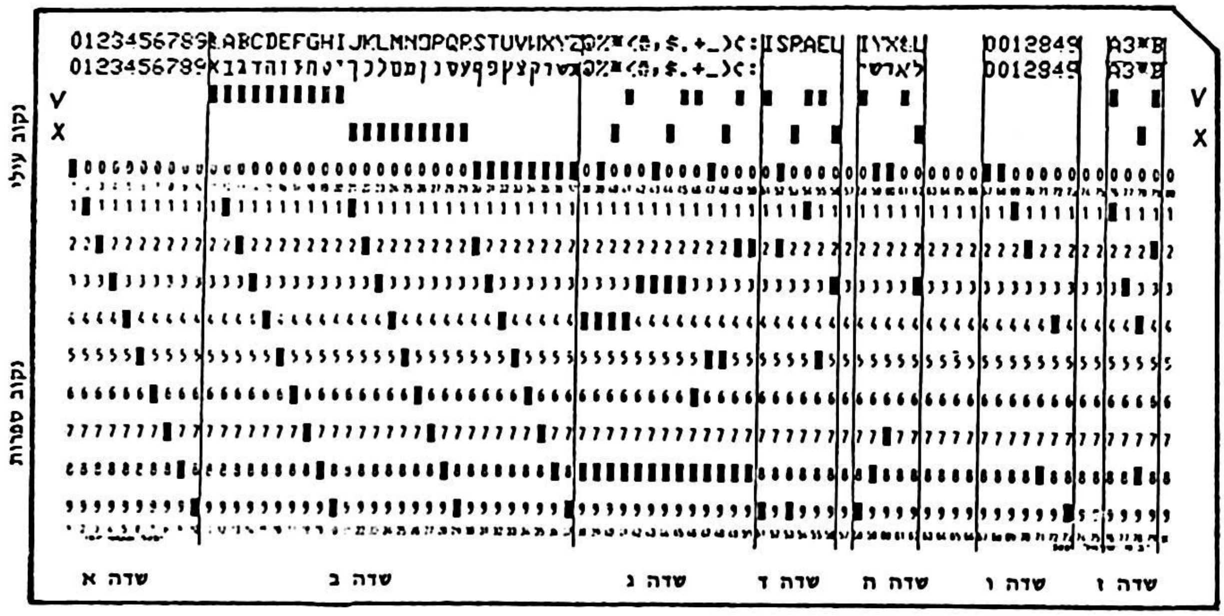
4.1 הכרטיס המנוקב

האמצעי הנפוץ ביותר להצגת הנתונים לקראת תהליך העיבוד הינו הכרטיס המנוקב (ראה ציור 4-2).

הכרטיס המנוקב ממלא שני תפקידים. הוא משמש כאמצעי קלט למחשב והוא גם אמצעי לאחסון נתונים. ניתן לשמור קובץ כרטיסים למשך תקופה ארוכה ולעבד אותו בדרכים שונות.

בפרק זה נעסוק בתכונת הכרטיס כאמצעי קלט, מכיוון שהוא איבד את תפקידו כאמצעי אחסון, כאשר הוכנסו לשימוש אמצעי האחסון המגנטיים, כפי שיוסבר בהמשך. היום קשה למדי למצוא את הכרטיס

כאמצעי קלט גם כן, מכיון שרוב הקלט החיצוני מגיע כעת למחשב באמצעות מסופים למיניהם. הדיון הינו בחלקו פרק בהיסטוריה של המחשוב.

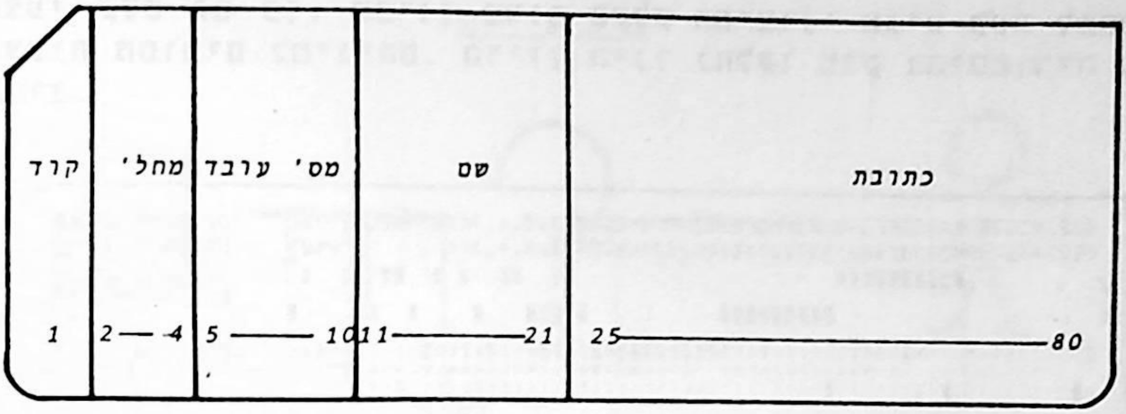


ציור 4-2 - שורות ועמודות על הכרטיס

הכרטיס מחולק ל-12 שורות ו-80 טורים (עמודות). מתוך 12 השורות, 10 השורות התחתונות מסומנות במספרים מ-0 ועד 9. הצגה של ספרות נעשית ע"י ניקוב חור בשורה המתאימה. אם נרצה להציג את הספרה 7 - ננקב חור בשורת ה-"7", הצגת הספרה 1 תיעשה ע"י ניקוב בשורת ה"אחד"-ים. אם רוצים לנקב מספר בעל 4 ספרות, נניח 1948 - נצטרך לנקב 4 נקבים: נקב ראשון (משמאל) בשורה של 1, נקב שני בשורת ה-9, נקב שלישי בשורת ה-4 ונקב רביעי בשורת ה-8. יש לשים לב, שארבעת הנקבים של "1948" חייבים להיות בטורים <u>צמודים</u> זה לזה (בשלב זה אין חשיבות אם אלה טורים 1-4 או 25-28, למשל).

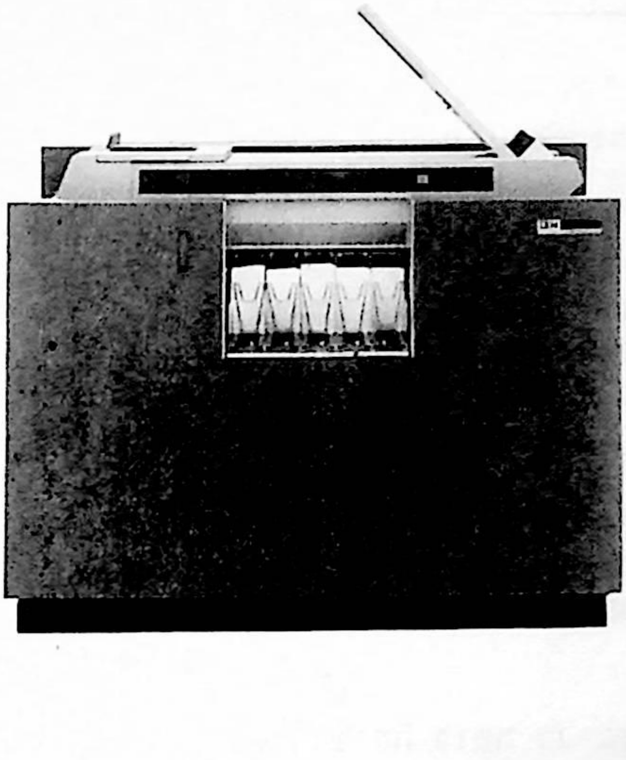
ניקוב של אותיות נעשה בשיטה דומה לניקוב ספרות, אך בהבדל אחד: כל אות מנוקבת על ידי שני נקבים בטור אחד, כאשר אחד הנקבים מופיע בחלק העליון של הכרטיס המסומן באזור ניקוב של אותיות (שורה 0, 11 או 12). כדי להבין את הטכניקה - נתבונן בציור 4-2.

פעולת הניקוב מתבצעת <u>במכונת ניקוב</u> (CARD PUNCH). המכונה דומה במעט למכונת כתיבה בכך, שיש לה לוח קלידים, אך היא מפיקה ניקובים בכרטיס, במקום הדפסת תווים. בציור 4-3 מוצגת רשומה של תושב תל-אביב, בשם דוד לוין. הרשומה, אשר מכילה את מס' תעודת הזהות, השם, הכתובת ומספר הטלפון של דוד לוין - נוקבה במכונת ניקוב של כרטיסים.



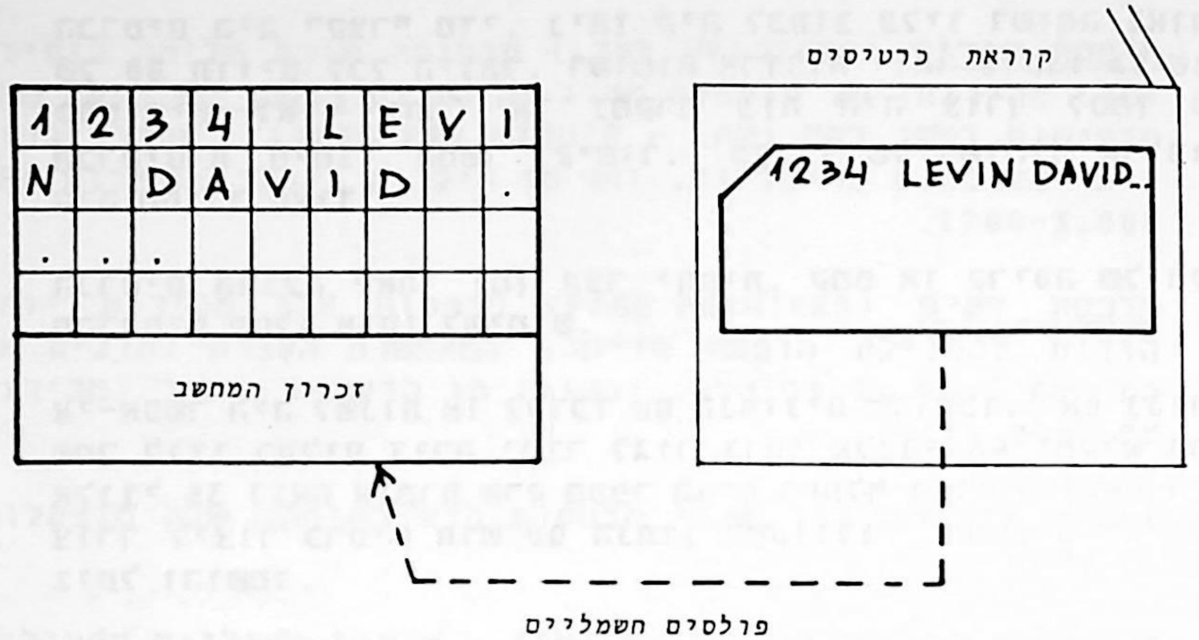
ציור 4-3 - רשומה

כדי שרשומת הנתונים, אשר מנוקבת בכרטיס, תעבור עיבוד במחשב, עליה להגיע אל יחידת הזיכרון שלו. זה ייעשה בעזרת מכונה הנקראת קוראת כרטיסים. נסביר בקצרה את אופן הפעולה של קוראת הכרטיסים של חברת יבמ.



ציור 4-4 -
קוראת כרטיסים

יחידה זו מחולקת לשני חלקים: החלק הימני נקרא <u>קוראת כרטיסים</u> (CARD READER), והשמאלי - <u>מנקבת כרטיסים</u> (CARD PUNCH).
קריאת כרטיסים בעזרת המכונה נעשית באופן הבא: הכרטיסים מוזנים דרך פתח ההזנה מצדה הימני של היחידה. הכרטיסים עוברים דרך תחנת קריאה בה הם פוגשים מברשות מתכת זעירות טעונות מתח חשמלי, אשר נוגעות בגליל מתכת. כאשר הכרטיס עובר בין המברשות לבין גליל המתכת - הוא מנתק את המעגל החשמלי. במקומות, שבהם הכרטיס מנוקב, נוגעות המברשות בגליל וסוגרות מעגל חשמלי. נוצר פולס חשמלי, אשר עובר מיחידת הקריאה אל יחידת הבקרה של המכונה, ומשם, בכבל אל היע"מ. מהירות הקריאה היא 1,000 כרטיסים בדקה, שהם 1,330 תווים בשנייה.



ציור 5-4 - העברה של תוכן הכרטיס אל הזיכרון

ביע"מ נמצאת יחידת הזיכרון, אשר קולטת את הנתונים מהכרטיס ומאחסנת אותם בבתים של הזיכרון. בכל בית יאוחסן תו אחד של הרשומה שנקראה. מה ייעשה בנתונים שזה עתה נקלטו? בזיכרון של המחשב צריכה להימצא תכנית, שתפקידה לטפל בהם. איך ומתי - על כך ידובר בפרק 5, בינתיים נניח שתכנית זו מצפה לנתונים ואכן יודעת מה לעשות בהם.

הניקוב ביחידה זו נעשה כפעולת פלט, כאשר נתונים הנמצאים בזיכרון המחשב מנוקבים בכרטיסים לשם העברה למתקן אחר, לשם עיבוד חוזר או למטרה אחרת כלשהי. הניקוב הוא אוטומטי ונעשה בעת מעבר הכרטיס במתקן הניקוב. מהירות הניקוב כ-250 כרטיסים בדקה.

הכרטיס המנוקב "שלט בכיפה" עשרות בשנים. חשבונות חשמל וטלפון נוקבו בכרטיסים, הודפסו ונשלחו לבתי הצרכנים בדואר. בבתי-חרושת ובמחסנים עברו לניהול מלאי ממוכן, וכרטיסי יבמ, שהונחו בקופסאות של הפריטים השונים במחסנים סימלו יותר מכל את המעבר מעיבוד נתונים ידני לענ"א - לעיבוד נתונים אוטומטי/אלקטרוני.

בתחילת הפרק הוזכר, שהכרטיס מילא תפקיד חשוב גם באחסון נתונים. היתרון החשוב היה בכך, שהוא אפשר עיבוד חוזר, במכונות, של הנתונים שאוחסנו, ללא צורך בניקוב מחדש. בקובצי כרטיסים נאגר מידע רב במקומות העיבוד השונים. קובצי תושבים בעירייה, או קובצי פריטים במחסן הוקמו ועובדו כקובצי כרטיסים. במשך הזמן התגלו החסרונות של הכרטיס המנוקב כאמצעי אחסון והוא נדחק מפני אמצעי האחסון המגנטיים. נמנה כמה מהבעיות, שדרשו פתרון:

* הכרטיס היה "קצר" מדי. ניתן היה לכתוב עליו רשומה באורך של 80 תווים לכל היותר. רשומות ארוכות יותר נוקבו על שני כרטיסים (או יותר) אך במקרה כזה היה צורך לסמן על הכרטיסים סימני המשך וקישור. עיבוד של רשומות ארוכות היה מסובך מאוד.

* הכרטיס התבלה לאחר זמן קצר יחסית. קמט או קריעה של חלק מהכרטיס פסלו אותו לשימוש.

* אי-אפשר היה לשנות או לעדכן את הנתונים שנוקבו. אם נניח, שמר לוין החליף דירה ועבר לגור ברח' אלנבי 48 במקום רח' אלנבי 36 (זאת אומרת שרק מספר הבית התחלף בשדה כתובת) היה צורך ליצור כרטיס חדש עם הנתון המעודכן. הכרטיס "הישן" בוטל והושמד.

* הטיפול בקבצים גדולים יחסית הפך למסורבל. קובץ התושבים בעיר כמו תל-אביב הכיל כ-300,000 כרטיסים (כמספר התושבים). לאחזקת קובץ כזה נדרשו כ-150 מגירות כאשר בכל מגירה נמצאים 2,000 כרטיסים. איתור כרטיס מסוים והחזרתו למקום לאחר הטיפול בו גזלו זמן רב, זמן הרבה יותר ארוך, מאשר טיפול בתוכן הכרטיס עצמו.

* מהירות העיבוד של הכרטיסים היתה נמוכה בהשוואה לכמויות הגדולות שצריך היה לעבד.

4.2 מדפסות

<u>מדפסת</u> (PRINTER) הינה יחידת פלט, אשר מאפשרת לראות פלט מעובד בצורה מודפסת ונוחה לשימוש. מהירות ההדפסה נעה בין 15 תווים בשנייה (במדפסת איטית) עד 68,000 תווים בשנייה (מדפסת 3800 של חברת יבמ, הפועלת עם קרן לייזר).

כיחידת פלט, מקבלת המדפסת את הנתונים בצורת אותות חשמליים ממערכת המחשב. סימנים אלקטרוניים אלה נקלטים במנגנון המדפסת, ובהתאם לסוג המדפסת הם הופכים לתווים, שמודפסים על נייר רציף. השימוש בנייר רציף מאפשר הדפסה בקצב וברציפות, ללא התערבות המפעיל.

ניתן לסווג את המדפסות בדרכים שונות. לפי צורת ההדפסה ומהירותה מבחינים בשלושה סוגים:

* <u>מדפסת סריאלית</u> (SERIAL PRINTER), אשר מדפיסה תו אחר תו, ולכן מהירות ההדפסה נמדדת <u>בתווים לשנייה</u> (CHARACTERS PER SECOND - CPS). מדפסות אלה הן איטיות וזולות בדרך כלל (מהירות 15-600 CPS).

- **מדפסת שורות** (LINE PRINTERS) מדפיסה שורה שלימה ביחידת זמן. מהירותן של מדפסות מסוג זה נמדדת **במספר השורות**, שהן מדפיסות **במשך דקה** (LINES PER MINUTE - LPM). מחירן גבוה יותר ממדפסות סריאליות, ואף מן היקרות ביותר (מהירות LPM 200-3,000).

- **מדפסת דפים** (PAGE PRINTERS) מדפסות אלו מהירות יותר הודות לטכניקת הדפסה חדישה, המאפשרת העברת נתונים אל המדפסת בנפח של דף שלם, וכמובן הן היקרות ביותר (מהירות 21,000 LPM).

דרך נוספת להבחנה בין סוגי מדפסות היא השיטה, שבה מופעלות יחידות אלה:

- **מדפסות הקשה** (IMPACT PRINTERS) - אותות חשמליים מפעילים מנגנוני הדפסה, אשר נלחצים אל הנייר. לקבוצה זו שייכות מדפסות **השרשרת** (CHAIN), מדפסות **חגורה** (BELT), מדפסת **נקודות** (DOT-MATRIX), מדפסות **כדוריות** (BALL) ומדפסות **מניפה** (DAISY WHEEL).

בכל המדפסות האלה (פרט למדפסת נקודות) צורת האות המודפסת יצוקה באלמנט ההדפסה וקבועה מבחינת הצורה והגודל. האלמנטים נעים לקראת הנייר בזמן ההדפסות ונלחצים אליו בתיווך של סרט ספוג דיו. בגלל תנועת האלמנטים, המלווה בחיכוך, מהירות המדפסות מוגבלת. במדפסת נקודות נוצרת צורת התו באופן דינמי ע"י קבוצת "מחטים" מתוך מטריצה נתונה (7x9 מחטים, למשל).

- **מדפסות ללא הקשה** (NON-IMPACT PRINTER) - הסימן המודפס לא נקבע מראש, אלא "נצרב" לתוך הנייר בעזרת חום או בתוך כדי תהליכים כימיים או פוטו-אלקטריים.



ציור 4-5.1 - מדפסת נקודות של דיגיטל

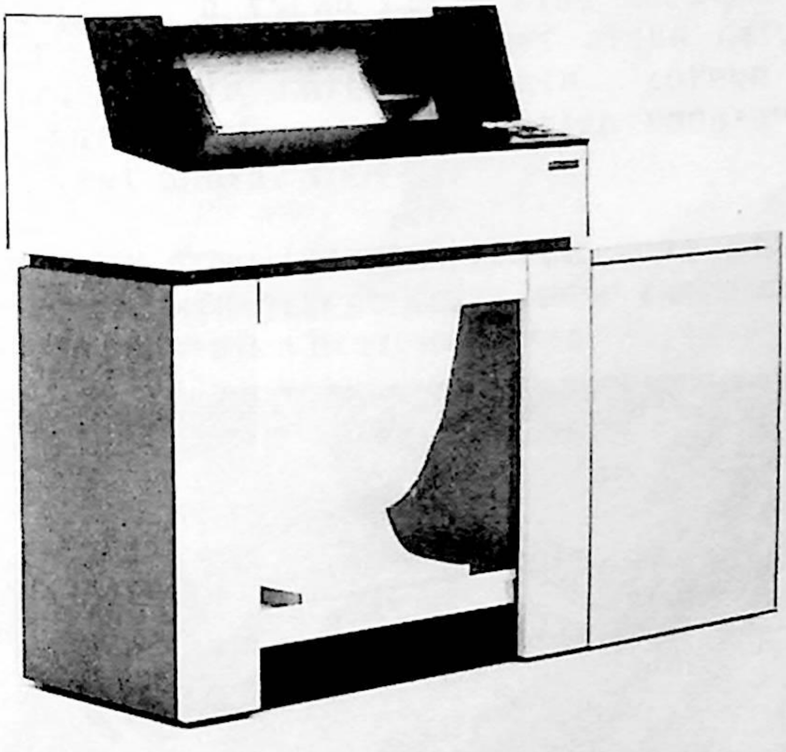
נציג כאן שני סוגי מדפסות: האחת נפוצה מאוד, בעלת טכניקת הדפסה של "שרשרת", והשנייה - מדפסת חדישה בעלת קרן לייזר.

1. <u>מדפסת שרשרת</u>

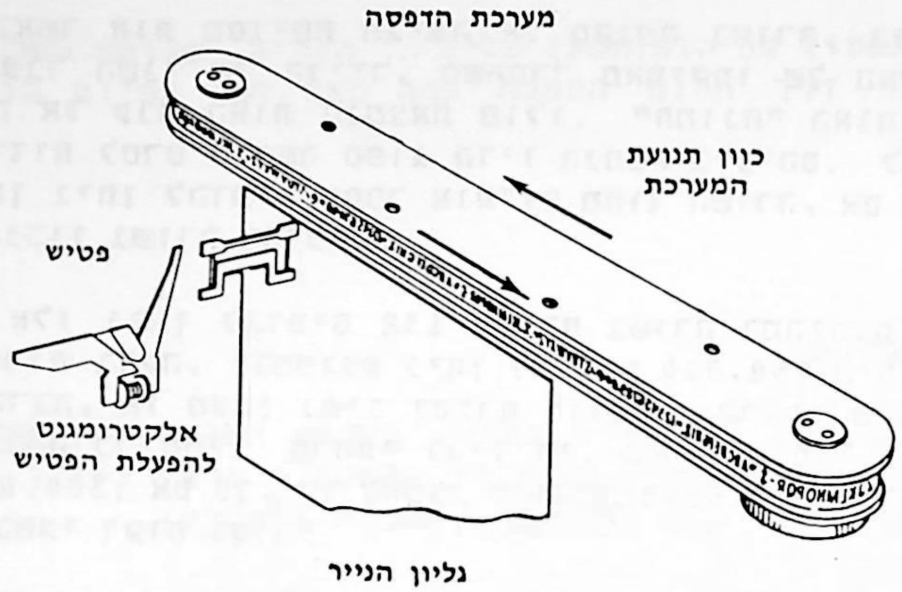
את תכונות המדפסת הזו נציג בעזרת המדפסת 3203 של יבמ.

בחזית המדפסת, שגובהה כ-1.2 מטר, מבחינים בפתח להכנסת נייר, ובפתח שדרכו יוצא הנייר לאחר הדפסה. בין שני הפתחים נמצאת מערכת הדפסה, שלה שלושה חלקים עיקריים:

- גלגלי שיניים מובילים
- שרשרת הדפסה
- מנגנון הדפסה



ציור 6-4 - מדפסת 3203 של חברת יבמ





חוליות „שרשרת" במתקן ההדפסה

ציור 7-4 - שרשרת הדפסה

תפקיד <u>גלגלי השיניים המובילים</u> ("טרקטורים") הוא להזין את הנייר בזמן ההדפסה. הם נמצאים מעל הקצוות של הפתח התחתון, ומסתובבים בכיוון מלמטה למעלה, כך ששיני הגלגלים נכנסים אל החורים שבשולי הנייר, ומעלים אותו כלפי מעלה.

<u>שרשרת ההדפסה</u> הינה שרשרת מתכת, שעליה נמצאים כל הסימנים הדרושים להדפסה. כמה סימנים דרושים לנו? באל"ף-בי"ת העברי יש 27 אותיות, בנוסף לכך יש צורך בעשרה סימנים בעבור הספרות מ-0 עד 9, כלומר בסה"כ 37 סימנים. דרושים עוד כ-11 סימני כתיבה שונים כגון נקודה, סימן שאלה, פסיק, סימן לחיבור, לכפל וכו', ולכן מערכת של 48 סימנים תספיק לנו. על השרשרת נמצאות 5 מערכות סימנים, בצורה של תבליט מיוחד לכל סימן. השרשרת מסתובבת במהירות על שני צירים הנמצאים במערכת ההדפסה.

<u>מנגנון ההדפסה</u> מורכב ממערכת פטישים זעירים, מנגנון אלקטרומגנטי אשר מפעיל אותם, ומסרט בד רחב ספוג בדיו.

<u>תהליך ההדפסה</u>

התבנית של השורה המודפסת נשלחת מן המחשב אל המדפסת, והמנגנון האלקטרוני שלה מפענח את השדר הזה. שרשרת ההדפסה מסתובבת כל

הזמן, וכאשר אות מסוימת מגיעה אל מקומה בשורה, הפטיש, אשר נמצא מהעבר השני של הנייר, משתחרר מאחזקתו של האלקטרומגנט שלו ומכה אל עבר האות הנמצאת מולו. "תמונת" האות נשארת על הנייר הודות לסרט הדפסה ספוג הדיו הנמצא ביניהם. למעשה, בכל יחידת זמן ניתן להדפיס מספר אותיות מתוך השורה, אם הן נמצאות במקומן הנכון בשורת ההדפסה.

במדפסות אלו ניתן להדפיס 132 תווים בשורה במהירות הדפסה של 1,200 שורות בדקה, ובממוצע ניתן להדפיס 150,000 סימנים בדקה. האם זה הרבה, או מעט? נשיב בעזרת דוגמה. קראתי ספר כיס משל הסופרת אגתה כריסטי. טרחתי וגיליתי, שמספר סימני הדפוס בספר הוא כ-300,000. אם כך, קל לחשב, שמדפסת 3203 יכולה להדפיס ספר כיס כזה בשתי דקות בערך.

מדפסות שרשרת יכולות לפעול במהירות הדפסה גבוהה, עד 3,500 שורות בדקה.

סוגים שונים של שרשרת

שרשרת ההדפסה, המוצגת בציור 7-4 הינה דוגמה לשרשרת "אנגלית". בדומה לה מצויות גם שרשרות "עבריות" או שרשרות "דואליות". באמצעות שרשרת דואלית אפשר להדפיס בעברית ובשפה לועזית בעת ובעונה אחת. ניתן גם להזמין שרשרת עם סימנים מיוחדים. החלפת שרשרת אחת באחרת אינה פעולה מסובכת, ומפעיל של מחשב מבצע זאת בפחות מדקה.

2. מדפסת לייזר

מערכת ההדפסה 3800 של חברת יבמ נבדלת מהמדפסת 3203 בשני תחומים עיקריים: זוהי מדפסת אלקטרו-גרפית, הפועלת ללא הקשה. פעולתה מתבססת על שימוש בקרני לייזר, והיא שייכת לקבוצה של "מדפסות דפים".

אופן הפעולה

המחשב שולח אל המדפסת שורות מוכנות להדפסה. השורות מאוחסנות בזיכרון המדפסת, שגודלו 52K, עד להשלמתו של דף הדפסה שלם. כאשר הדף מוכן להדפסה, הוא מוקרן ע"י קרני לייזר על גבי תוף גדול, המסתובב כל העת. לתוף זה מעטפת, בעלת תכונות מוליכות-צילומיות. המעטפת נטענת בתחילת הסיבוב במטען חשמלי חיובי, אשר מתפרק במקומות, שבהם פוגעת קרן הלייזר. בצורה כזו מתקבל על מעטפת התוף התשליל (נגטיב) של הדף השלם, כאשר הוא בעל מטען שלילי.

גרגירי דיו שחורה מיובשת, המכוסים שעוה וטעונים מטען חיובי, נמשכים אל התשליל שעל המעטפת המסתובבת. כאשר התמונה "החיובית" הזו מגיעה אל "עמדת ההדפסה", היא מועברת אל נייר

ההדפסה, הטעון גם הוא מטען שלילי. הנייר, שנושא עתה את תמונת הדף, עובר דרך תחנת הספגה חמה מאד לשם ספיגת הדיו וסיום התהליך.

תכונות המערכת

- שיטת ההדפסה, שתוארה לעיל, נקראת השיטה האלקטרו-גרפית. בשיטה זו משורטטות האותיות באמצעות קרני לייזר. האותיות במדפסת זו ניתנות לתכנות ונשמרות בזיכרון.

- ניתן לתכנת לא רק אותיות, אלא גם צורות גרפיות שונות.

- אפשר להדפיס בשורה בין 136 ל-204 אותיות, תלוי בצפיפות האותיות.

- אפשר להדפיס על דף ההדפסה בין 60 ל-120 שורות, תלוי בצפיפות השורות.

תכונות אלו גורמות ליתרונות הבאים:

- מהירות ההדפסה גבוהה. למשל, בהדפסה בצפיפות רגילה (6 שורות לאינץ') מתקבלת במדפסת 3800 מהירות אופטימלית של כ-10,000 שורות בדקה ויותר, בהשוואה למהירות של כ-1,200 שורות במדפסת 3203, או כ-3,000 שורות במדפסות הקשה מהירות. אפשר להדפיס עד 20,000 שורות בדקה, שהם כ-500 עמודים.

- מהירות ההדפסה קבועה, לפי דפים - אינה תלויה במבחר התווים (בהשוואה למספר הופעות התווים בשרשרת ב-3203) ואינה תלויה בדילוגי שורות בדף.

- ההדפסה היא ברורה, מיקום האותיות מדויק, אין פטישים פגומים ואין הזזה של האותיות.

- ניתן לקבל בקלות כל צירוף אותיות, ע"י הפעלת תכנית שירות מיוחדת. אין צורך בשרשראות הדפסה מיוחדות. כל זה פותח אפשרויות חדשות ורבות בתכנון הפלט - סימנים מיוחדים, אותיות גדולות (BOLD FACE), שרטוט גרפים, שרטוט תוי נגינה, ועוד.

- במדפסות החדישות מסוג זה ניתן לשלב תמליל ושרטוטים המגיעים מקבצים שונים.

4.3 הסרט המגנטי

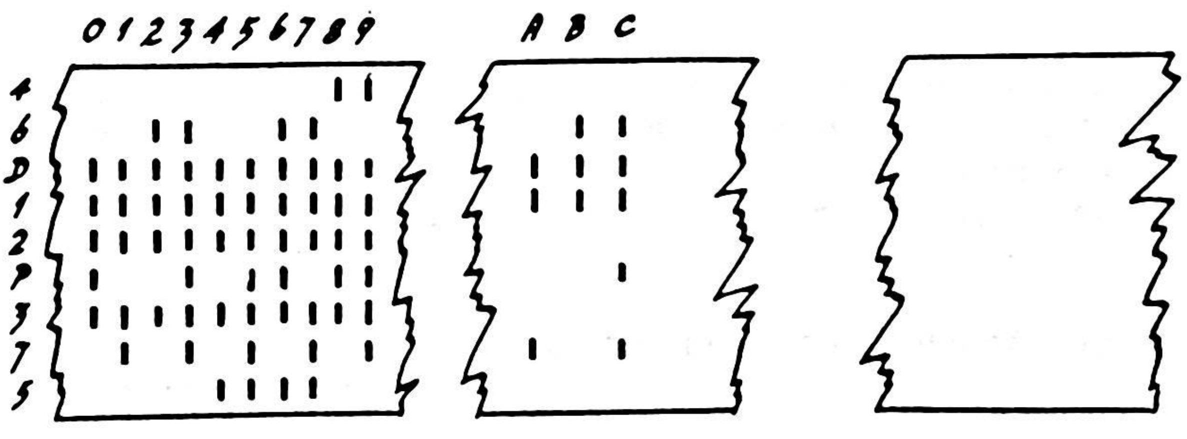
חיי הכלכלה של מדינות מפותחות ומתפתחות באמצע המאה העשרים הצטיינו בתכונות הבאות:

- התפתחות טכנולוגית ניכרת
- מחזור כספים גדול מאוד
- התפשטות העיסקות על פני אזורים נרחבים או אפילו ארצות רבות
- תחרות חזקה

בתנאי תחרות חזקה חשוב להשיג את המידע הדרוש לקבלת החלטות מהר ככל האפשר. התפשטות העסקות מבחינה גיאוגרפית והמחזור הגדול חייבו אחזקת נתונים בהיקף הולך וגדל ועיבוד הנתונים היה צריך להתאים את עצמו לתנאים אלו. כל אלה, והמגבלות של הכרטיס, עודדו פיתוח טכנולוגי גם בשטח עיבוד הנתונים, וכך הופיע הסרט המגנטי.

הסרט המגנטי הוא אמצעי קלט ופלט וגם אמצעי לאחסון נתונים. הוא עונה על מרבית הצרכים של חיי הכלכלה המודרניים בתחום של עיבוד הנתונים. גם הוא נדחק ממעמדו החשוב, ככל שהתפתחו שיטות עיבוד מקוונות, ואת מקומו תפס הדיסק המגנטי. בהמשך נתאר את הסרט הנפוץ, הארוז בגלגל בקוטר של 10 אינטש, שרוחבו חצי אינטש ובו 9 ערוצי מגנוט. יבמ הכריזה לאחרונה סרטים מגנטיים חדשים הארוזים בקלטות, ולא נדון בכך כאן.

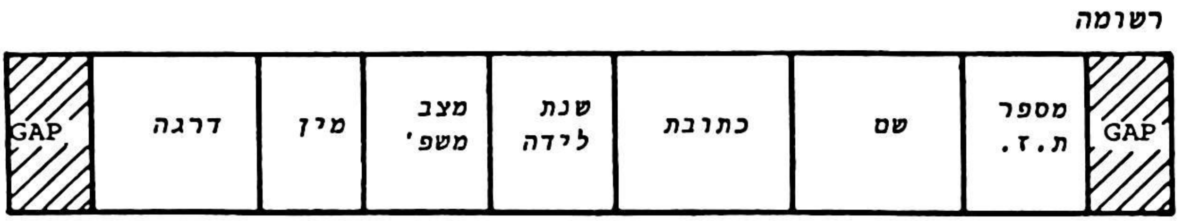
הסרט המגנטי (בדומה לסרט של רשמקול או של קסטות) עשוי מחומר פלסטי גמיש, שעליו ציפוי דק, אשר ניתן למגנוט. על סרט, שרוחבו כסנטימטר (חצי אינטש) ישנם 9 (או 7 סוגי סרטים מסוימים) שורות, או ערוצי מגנוט. כתיבה או "הקלטה" על הסרט וקריאה ממנו, נעשים באמצעות ראש קריאה וכתיבה, אשר מותקן ביחידת הסרט המגנטי. על הסרט ניתן להציג אותיות, ספרות, או סימנים מיוחדים, כפי שרואים זאת בציור 4-8. הנתונים רשומים על הסרט בצפיפות גדולה מאוד. על הסרט המגנטי הסטנדרטי של חברת יבמ, למשל, ניתן לכתוב 6,250 תווים לאינטש אחד.



ציור 4-8 - הצגת סימנים על סרט מגנטי

הסרט המגנטי כאמצעי אחסון נתונים מצטיין בכל אותם תחומים בהם נתגלו חסרונות הכרטיס המנוקב:

- רשומה הנכתבת על סרט מגנטי, אינה מוגבלת באורכה, והיא יכולה להכיל מאות ואפילו אלפי תווים.



ציור 4-9 - רשומה על סרט מגנטי

- הסרט עשוי מחומר, אשר עומד היטב בתלאות הזמן והשימוש, ואינו מתבלה אלא לאחר שנים.

- ניתן להשתמש בסרט שימוש חוזר.

- הטיפול בסרטים מגנטיים נוח מאוד בגלל הנפח הקטן, שהם תופסים. סרט בעל צפיפות רגילה יכול להכיל 30 מליון תווים, וסרט בעל צפיפות גדולה - כ-100 מליון! אם נזכור, שבספר כים ממוצע יש כ-300,000 סימני כתיבה, הרי סרט "בינוני" יכיל "ספרייה" בת 100 ספרים, והסרט השני כ-300 ספרים כאלה.

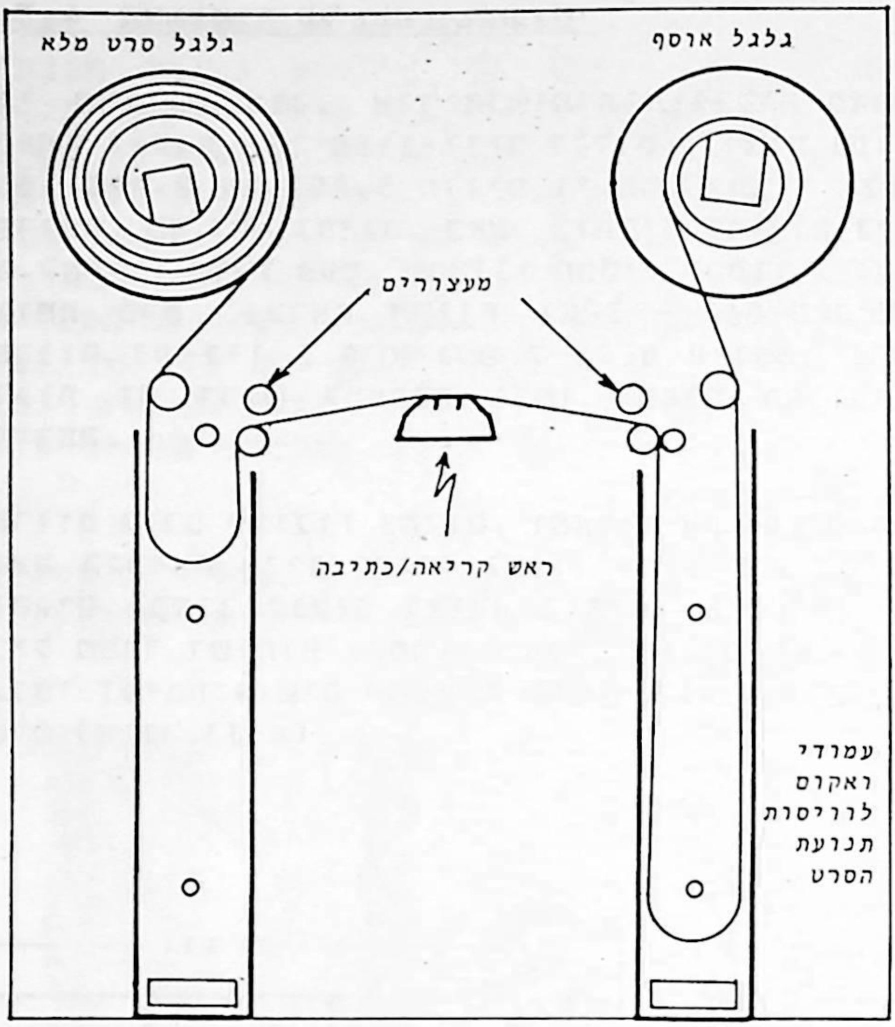
- מהירות הקריאה והכתיבה בסרט מגנטי יכולה להגיע למיליון תווים בשניה ויותר, בהתאם לצפיפות המיגנוט ומהירות הפעולה של כונן הסרט.



ציור 10-4 - כונן סרט מגנטי

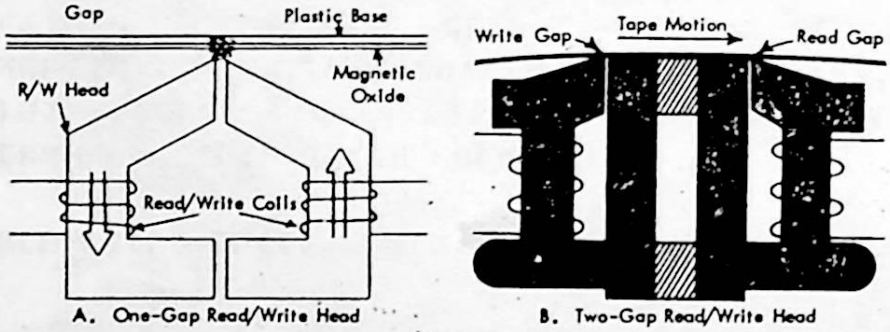
4.3.1 קריאה וכתיבה בסרטים מגנטיים

כתיבה או קריאה של סרטים מגנטיים נעשית בכוננים מיוחדים המיועדים לכך (ציור 10-4). על הכונן נמצאים שני גלגלים, אשר מסתובבים בעזרת שני מנועי חשמל נפרדים. בין שני הגלגלים נמצאת מערכת המכילה את ראש הקריאה/כתיבה (ציור 11-4). בגלגל אחד נמצא הסרט, שממנו קוראים או שמתכוונים לכתוב עליו והגלגל השני ריק.



ציור 4-11 - ראשי קריאה/כתיבה

פעולת הכתיבה יכולה להתחיל על פי פקודה מתכנית המחשב. הסרט מתחיל לנוע מגלגל אחד לשני. באותו זמן נשלחים אותות חשמליים מן המחשב (בהתאם לתוכן תאי הזיכרון שרוצים לכתוב על הסרט). זרם החשמל, המגיע אל הסליל, שבראש הכתיבה הופך לשדה מגנטי וממגנט את משטח הציפוי (ציור 4-12, חלק שמאל). תנועת הסרט המהירה וקצב המיגנוט מאפשרים לכתוב על הסרט במהירות גבוהה מאוד. הכתיבה על הסרט המגנטי הורסת את התוכן, שהיה כתוב בו קודם.



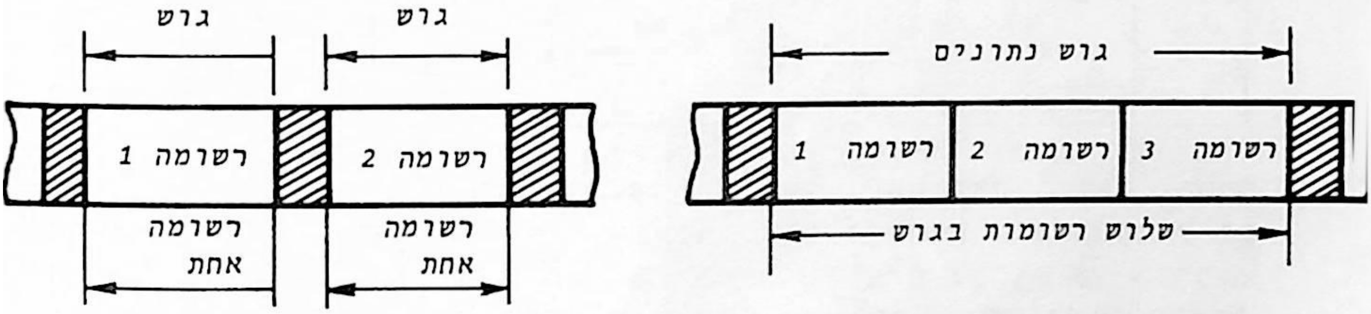
ציור 4-12 - כתיבה על סרט

קריאה מסרט מגנטי מתבצעת כאשר הסרט נע מגלגל אחד לשני, ועובר מתחת לראש הקריאה. כתוצאה מכך נוצרים אותות חשמליים בסליל של הראש, אשר מועברים אל יחידת הבקרה של הכונן, ומשם בכבל אל הזיכרון של המחשב. קריאה מסרט אינה הורסת את הנתונים שרשומים בו, וניתן לקרוא ממנו פעמים רבות.

4.3.2 הרשומות על הסרט המגנטי

כפי שהודגש כבר, אין הרשומות מוגבלות מבחינת האורך, אולם מטעמי נוחות אנו מעוניינים לקלוט קבוצות נתונים גדולות, נניח 500, 2,000 או 5,000 תווים ויותר, לעצור ולעבד אותם ולהמשיך בקריאה. כך גם בכתיבה, כאשר כותבים קבוצות נתונים בזה אחר זה. יש לציין, שבכל פעם, שהכונן מפסיק לכתוב על הסרט - נוצרת מעין רשומת סרט הנקראת __מרווח__ (INTER RECORD GAP - IRG). גודל המרווח נע בין 0.3 אינטש ל-0.75 אינטש, בהתאם לסוג הכונן. מרווח זה דרוש למערכת בזמן התחלת התנועה של הסרט ובזמן עצירתה.

המרווח גורם לבזבוז במקום, ומאריך את זמן הקריאה של הקובץ, כי בשעת הקריאה חייב הכונן לעצור בכל פעם, כשיגיע למרווח. לכן נוהגים לכתוב קבצים בצורת __גושים__ (BLOCKS), כאשר כל גוש כזה מכיל מספר רשומות והמרווח נכתב רק בסופו. כתיבת קובץ על סרט מגנטי בשיטת גושים מאפשרת לחסוך גם בזמן העיבוד וגם במקום על הסרט (ציור 4-13).



ציור 4-13 - גושים (BLOCKS) של נתונים

מול יתרונותיו הרבים של הסרט המגנטי יש להזכיר גם חיסרון אחד: הגישה אל רשומות הקובץ הינה __סדרתית__ (SEQUENTIAL) בלבד. סדר הרשומות בקובץ נשאר קבוע, לפי סדר הכתיבה. קריאת הרשומה האחרונה מתוך הקובץ מחייבת מעבר על כל הקובץ, מראשיתו ועד סופו. במקרה של קובץ גדול מאד, הדבר עלול להמשך 3 עד 4 דקות (בהתאם למהירות תנועת הסרט בכונן).

הסבר מפורט למבנה הרשומות על סרטים מגנטיים - ראה בפרק 9.

4.4 התקליטון

אחד מהאמצעים לקליטת נתונים ואחסונם הוא __התקליטון__ (DISKETTE). הוא גם משמש יחידת קלט/פלט ויחידת אחסון חשובה, ולעתים בלבדית, של המיקרו-מחשבים.

התקליטון עשוי מחומר פלסטי מיוחד, שעליו ציפוי, שניתן למיגנוט, ועל כן מאפשר לרשום בו נתונים בצורה מגנטית. התקליטון, בדומה לתקליט רגיל, מסתובב על ציר במהירות קבועה. הנתונים נרשמים בו במעגלים קונצנטריים הנקראים **מסילות** (TRACKS). לתקליטון המקובל במחשבים, קוטר של 8 אינטש יש לו משטח מיגנוט אחד בלבד, אשר מכיל 77 מסילות, מתוכן מסילת אינדקס אחד ושלוש מסילות רזרבה. בכל מסילה אפשר לכתוב 26 רשומות באורך 128 תווים - ומכאן ניתן לחשב, שקיבולת התקליטון היא כ-240,000 תווים. זוהי תכולה, המקבילה לזו של 3,000 כרטיסים מלאים. תקליטון משוכלל יותר בעל צפיפות כפולה ואפשרות כתיבה משני צידיו יכיל קרוב למליון תווים (יותר מ-12,000 כרטיסים מלאים!).

כיצד כותבים על תקליטון? בציור 4-14 מוצגת התקלידית 3740 של יבמ. התקליטון מוכנס לאחת המגירות בקצה הרחוק של השולחן, ואז אפשר לכתוב עליו בעזרת המקשים של המקלדת.



ציור 4-14 - תקלידית ומדפסת

תהליך הכתיבה נעשה בצורה הבאה: לוחצים למשל על הקליד עם האות A. האות A תופיע על מסך התצוגה הקטן, שנמצא בצידו השמאלי של שולחן העבודה, ובאותו זמן הסימן A ירשם גם **במאגר** (BUFFER). ניתן להמשיך ולכתוב רשומה עד 128 תווים. כאשר מסתיימת כתיבת הרשומה רואים עדיין את כולה על המסך. ניתן לבדוק, או אפילו לתקן את הכתוב. לחיצה על מקש ENTER גורמת להעתקת הרשומה מן המאגר אל התקליטון. הכתיבה מתבצעת על מסילות 2 עד 74. מסילה מס' 1 היא מסילת אינדקס, בה נרשם המיקום של כל רשומה, שנכתבת על התקליטון.

התקלידית, כפי שהוצגה כאן, היא יחידה, אשר אינה מחוברת אל המחשב, והיא מיועדת לקליטת נתונים מטופס כלשהו אל התקליטון. התקלידית היא מכונה משוכללת הרבה יותר ממנקבת כרטיסים, כי היא כוללת יחידת עיבוד, אשר מסייעת לקלדנית בעבודתה על ידי הדרכה ובדיקה של הנתונים בעת ההקשה. בגלל העובדה, שהכתיבה כאן היא למעשה רישום מגנטי, ולכן ניתן לתקן או לעדכן את הכתוב על התקליטון. בעזרת האינדקס מאתרים את הרשומה המסוימת, בה רוצים לטפל, והרישום הודש מוחק באופן אוטומטי את התו אי השדה הישנים, וגורם לכתיבת הנתון החדש באותו מקום. ישנו מודל של תקלידית, שמתחבר למחשב בתקשורת.

יחידת תקליטונים 3540 של יבמ היא יחידת קלט/פלט, אשר מחוברת ישירות למחשב ומאפשרת קריאה/כתיבה של תקליטונים בעזרת תכנית מתאימה הנמצאת בזיכרון המחשב. מהירות הקריאה היא מעל 700 תווים לשנייה ומהירות הכתיבה בערך כ-500 תווים בשנייה. יחידה זו יכולה לקרוא תקליטונים, אשר הוכנו במערכות אחרות, כמו למשל במיקרו-מחשבים.

התקליטון החליף את הכרטיס המנוקב בגלל תכונת האחסון הגבוהה שלו, מהירות הקליטה למחשב, ובעיקר בגלל אפשרויות העיבוד של התקלידית כסיוע לקלדנית. ישנן תקלידיות, אשר משמשות כמערכות עיבוד נתונים פשוטות והן משתמשות בתקליטונים כאמצעי אחסון נתונים. ניתן לחבר לתקלידית מדפסת מטריצה איטית.

התקליטון הינו גם מוצר פלט של מיני ומיקרו מחשבים. במחשבים אלה מותקן כונן תקליטון, אחד או יותר, כיחידת קלט/פלט. הזנת הנתונים נעשית באמצעות לוח המקשים של המחשב, או של מסוף, המחובר אליו. הצגת הנתונים נעשית על המסך.

התקליטון נפוץ במערכות מחשבים לשמוש אישי, והוא משמש להם כיחידת האחסון העיקרית. במחשבים אלו קוטר התקליטון הוא 5 1/4 אינטש ויש לו משטח מגנוט אחד בתכולה של 180,000 תווים, או שני משטחי מיגנוט בתכולה כוללת של 360,000 תוים (PC של יבמ).

4.5 הדיסק המגנטי

בין מערכות הקלט והפלט ואמצעי האחסון, אשר נסקרים בפרק זה, נועד מקום מיוחד **לדיסק המגנטי** (MAGNETIC DISK), וזאת בגלל סיבות אחדות:

1. הדיסק המגנטי איננו אמצעי קלט/פלט במובן הרגיל של המלה, כי אין כאן העברה ישירה של נתונים מן "העולם החיצוני" אל הדיסק. אפשר ליצור קובצי נתונים על דיסק מגנטי, רק באמצעות מחשב ותכנית מתאימה.

2. הדיסק המגנטי הוא בעל תכונות פיסיות כאלה, אשר מאפשרות גישה ישירה אל הרשומה הבודדת מתוך הקובץ, הרשום עליו. לכן הוא גם נקרא **יחידת אחסון לגישה ישירה** (DIRECT ACCESS STORAGE DEVICE - DASD).

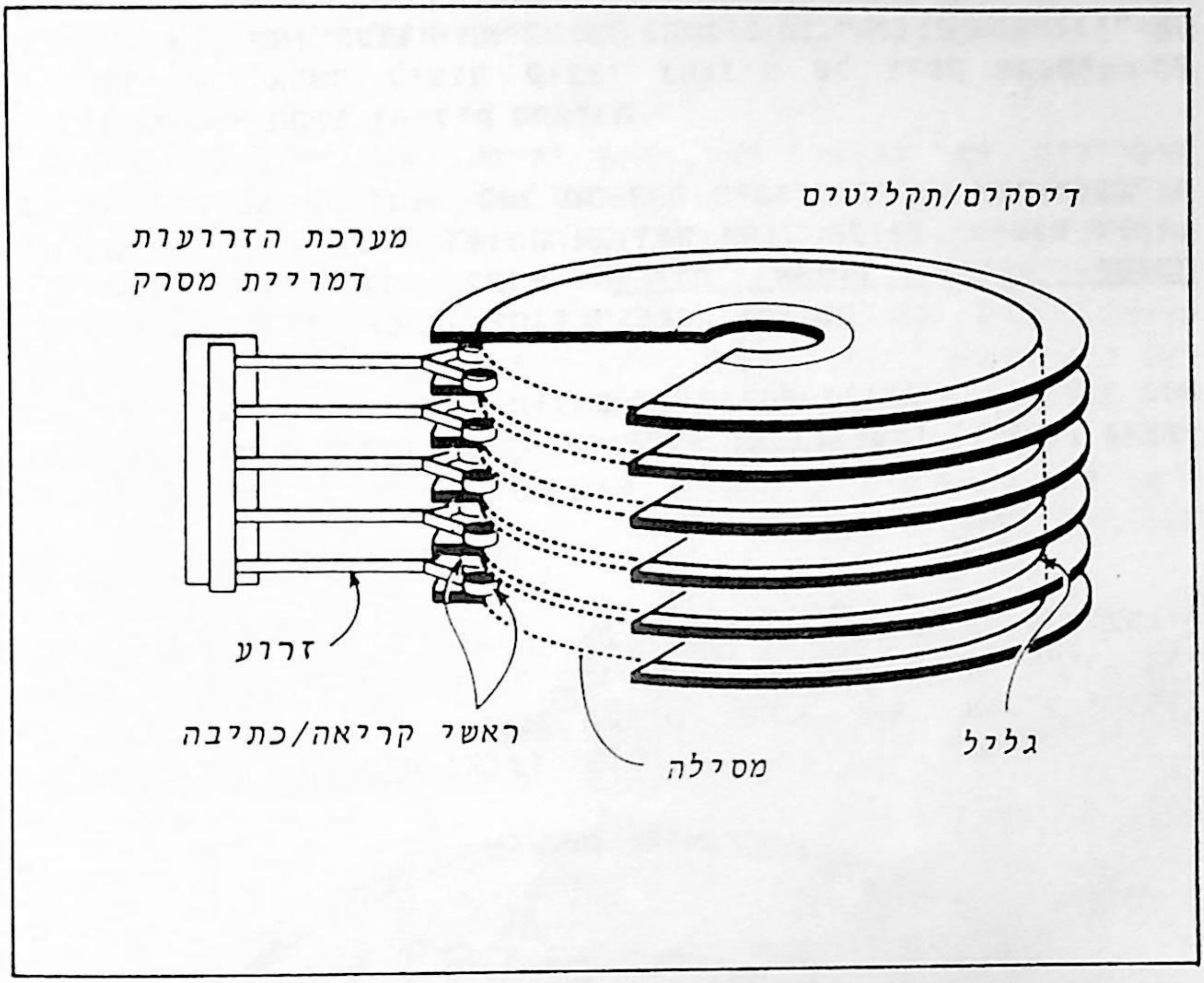
3. מהירות הקריאה של הנתונים מן הדיסק גבוהה מאד, ולכן הוא יכול לשמש **זיכרון עזר** (AUXILIARY STORAGE) לזיכרון הראשי של המחשב.

התכונות הפיסיות של הדיסק

הדיסק המגנטי מזכיר במבנהו את התקליטון, שנסקר בהרחבה בסעיף קודם. אך בעוד שהתקליטון קטן ובעל קיבולת נמוכה יחסית, הרי הדיסק המגנטי מורכב ממספר תקליטי מתכת, אשר יצוקים יחדיו, ומהווים **אגד תקליטים** (DISK PACK).



ציור 4-15 - דיסק מגנטי נייד



ציור 4-15.1 - אגד תקליטים ומנגנון קריאה/כתיבה

אגד התקליטים, או כפי שמקובל לומר - הדיסק, המוצג בציור 4-15 הינו דיסק נייד, כלומר, שניתן להסירו מהכונן ולהחליפו בדיסק אחר. לאחרונה מרבים להשתמש בדיסקים קבועים. בדיסק זה ישנם 11 תקליטים, ולכל תקליט שני משטחים ועליהם **המסילות** (TRACKS), שבדומה לתקליטון, מצופים חומר, שניתן למיגנוט. קוטר התקליט כ-"14, ומספר המסילות נע בין 200 ל-400, בהתאם למודל. על דיסקים מגנטיים ניתן לכתוב מ-100 מליון תווים ועד 3000 מליון תווים וניתן לקרוא במהירות של עד 3 מליון תווים בשניה, בהתאם למודל.

ראש קריאה/כתיבה

לכל משטח יש ראש קריאה/כתיבה, המיועד לבצע את הפעולות הללו. בדיסק שבציור 4-15.1 ישנם 6 תקליטים, שהם 12 משטחים. מכיון שהמשטח העליון ביותר והתחתון ביותר ניתנים לשריטה או לפגיעה פיסית, הם אינם בשמוש, ונותרים על כן 10 משטחים, שיש להם 10 ראשי קריאה/כתיבה. טיעון זה תקף עבור דיסק נייד, שניתן להסירו מהכונן ולהחליפו באחר. במערכת דיסקים קבועים משתמשים בכל המשטחים של הדיסקים, מכיון שהם מוגנים בתוך המערכת.

כל זוג ראשים מותקן על זרוע מיוחדת, וכל הזרועות יחד מחוברות לציר אנכי המותקן בתוך כונן הדיסקים. הציר יכול לנוע בצורה אופקית, ובכך להעביר את הראשים מן המסילה החיצונית של המשטח אל המסילה הפנימית ביותר. ראשי הקריאה/כתיבה "מרחפים" מעל משטח התקליט ואינם נוגעים בו.

פעולות קריאה וכתיבה

נניח עתה, שהמערכת קיבלה הוראה לקרוא רשומה מסוימת, הנמצאת על מסילה מס' 5 במשטח מס' 8.

מערכת הראשים נעה פנימה עד מסילה מס' 5, והפיקוח עובר לראש מס' 8. המשטח הממוגנט של הדיסק (המסתובב במהירות של עשרות סיבובים בשנייה) יוצר אותות חשמליים בראש הקריאה לפי הקטעים הממוגנטים בו. זרמים אלה מועברים אל יחידת הבקרה של הדיסקים, ומשם - אל הזיכרון של המחשב.

פעולת הכתיבה נעשית בעזרת אותם ראשים, אך כאן באים האותות החשמליים מן הזיכרון של המחשב, ופעולת הכתיבה מתבטאת במגנוט המשטח של המסילה. פעולת הכתיבה מלווה בציון המקום המדויק שבו נכתבת הרשומה.

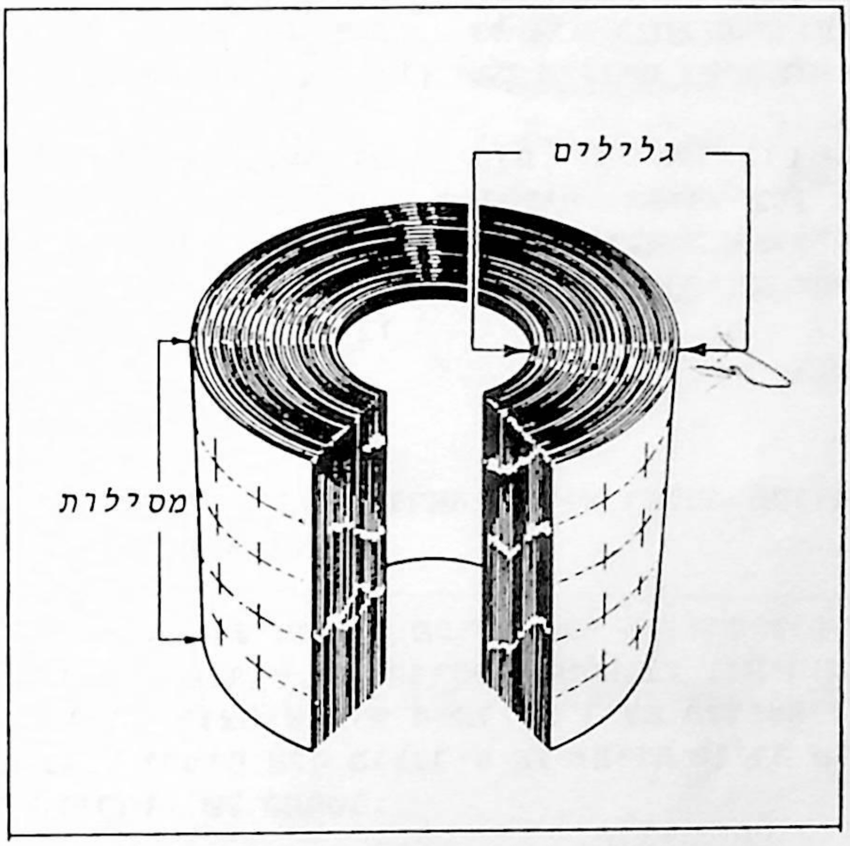
מושג הגליל

ניתן לכתוב על הדיסק המגנטי נתונים כמו על תקליטון, זאת אומרת, מסילה אחר מסילה. אופן כזה של כתיבה איננו יעיל די הצורך, כי הוא כרוך בהזזה מתמדת של הראשים, בעת מעבר ממסילה אחת לשנייה. כדי לייעל את הכתיבה על הדיסקים, תוכננה שיטה

חסכונית יותר - שיטת הגלילים. לפי שיטת כתיבה זו מתחילים לכתוב את קובץ הנתונים, נניח במסילה מס' 12 במשטח העליון. כאשר מסילה זו מתמלאת - ממשיכים לכתוב במסילה מס' 12 במשטח השני, אח"כ במסילה מס' 12 של המשטח השלישי, וכך הלאה. בצורה כזו חוסכים את הזזת המנגנון של הראשים, עד שמתמלאת המסילה אשר במשטח התחתון.

כל המסילות בעלות קוטר זהה יוצרות מעין <u>**גליל**</u> (CYLINDER), ראה ציור 4-16.

בדיסק בעל 200 המסילות ניתן ליצור 200 גלילים, שקוטר כל אחד מהם קטן יותר מחברו. יש לציין, שמספר התווים הנכתבים במסילות חיצוניות, זהה לזה שבמסילות הפנימיות. הדבר מאפשר מהירות סיבוב אחידה בזמן הפעולה.



ציור 4-16 גליל באגד תקליטים (דיסק)

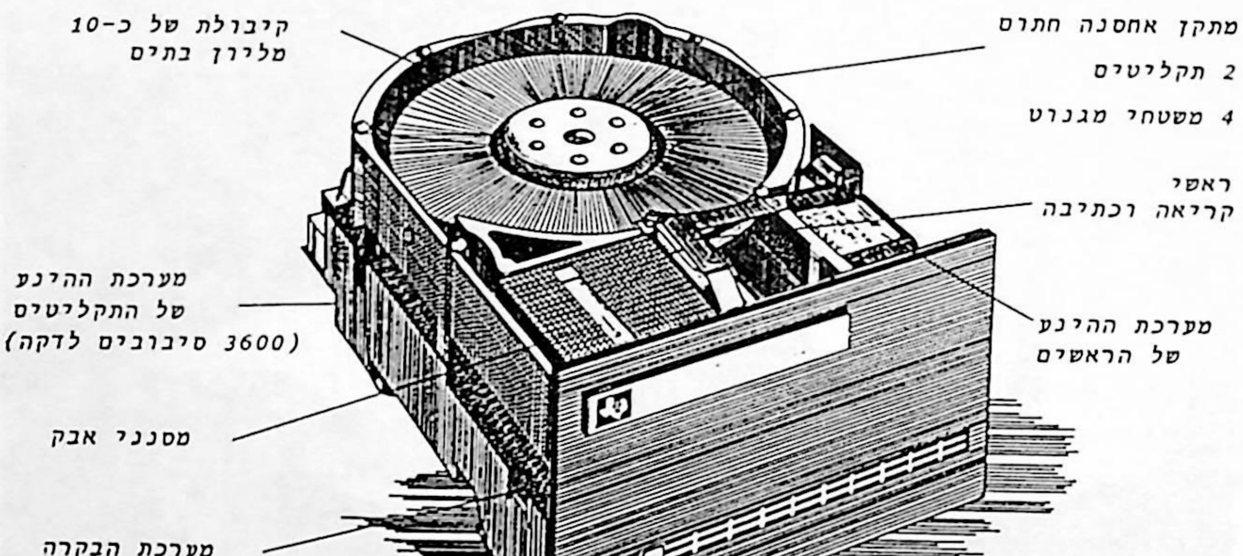
התכונות הפיסיות המיוחדות של הדיסק המגנטי מאפשרות עיבוד קבצים במהירויות גבוהות מאד. לדוגמה, הגישה אל רשומה בודדת נמשכת כ-20 אלפיות השניה בלבד. במשך שניה ניתן לקרוא או לכתוב על הדיסק עד 3 מליון בתים. למהירויות גבוהות כאלו משמעות מיוחדת בעיבודים מקוונים, עליהם ידובר בפרק 11.

הדיסק שתואר קודם מורכב על כונן הדיסקים בזמן העבודה, וניתן להחליפו באחר, אם יש צורך בכך. עובדה זו הינה בעלת משמעות רבה

לגבי מפעלים או ארגונים, שיש להם קבצים רבים וגדולים. הבה ניעזר בדוגמה: נניח, שקובץ הפריטים של כל המחסנים במפעל הינו בגודל של 2.5 מיליארד תווים. המפעל יכול להסתפק בארבעה כונני דיסקים לצורך עיבוד זה, כאשר על כל כונן מורכב אגד, המכיל 300 מליון תווים. כאשר נגמר הטיפול בחלק הראשון של הקובץ, ניתן להחליף את האגדים באלה, המכילים את חציו השני של הקובץ.

כאן המקום לציין, שסוג הדיסק המוצג לעיל איננו הסוג היחיד הקיים. במרוצת השנים נמצאו פתרונות טכנולוגיים מעניינים לכל הבעיות הכרוכות בשימוש בדיסקים. לדוגמה, בשנות השבעים הופיעו דיסקים עם **ראשי קריאה/כתיבה קבועים** (FIXED HEAD DISKS). בדיסקים הללו הוקצה ראש נפרד לכל מסילה, ולכן אפשר היה לחסוך את המנגנון המסובך של הזזת ראשי קריאה/כתיבה, ולהשיג מהירות גישה גבוהה יותר.

חידוש טכנולוגי אחר הינו יצירת יחידה שלימה, המכילה את הדיסקים ואת מנגנוני הראשים, ארוזים יחדיו. היחידה נפתחת אך ורק כאשר הדיסק מורכב במקומו על ציר הכונן. שמם הכולל של חידושים טכניים אלה הוא **טכנולוגיית וינצ'סטר** (WINCHESTER TECHOLOGY). דיסקים אלה מקובלים כיום, הן ביחידות מחשב גדולות, והן במיני ובמיקרו-מחשבים.



ציור 4-16.1 - דיסק וינצ'סטר עבור מיקרו-מחשב

4.6 מצג

יחידת קלט/פלט לשימוש בלתי אמצעי של בני אדם היא **יחידת תצוגה** (DISPLAY UNIT) או **מצג.** היא מקובלת כיום בתחומי שימוש ובמקומות שונים נמצא אותה ברוב מקומות העבודה, במשרד או בבית-החרושת, בבית-הספר או בבית (ציור 4-17).

היחידה מורכבת משני חלקים: **מקלדת** או **לוח מקשים** (KEYBOARD), אשר משמש כיחידת קלט, כאשר האדם מקיש עליו את הנתונים לשם הזנה למערכת, **ומסך** (SCREEN), הדומה למסך טלויזיה, שעליו מוקרן הפלט מן המחשב. על המסך מוצג הקלט, שמקישים במקלדת וגם הפלט, שנשלח מן הזיכרון של המחשב. רוב היחידות מצויידות **במאגר** (BUFFER) בהתאם ליכולת הקיבול של המסך. למסך של IIS, מתוצרת ישראל, קיבולת של 1920 תווים (24 שורות של 80 תווים כ"א). מסך 3278 או 3279 של יבמ יכולים להכיל 32 שורות (2560 תווים) או 43 שורות (3440 תווים), בהתאם למודל.



ציור 4-17 - מצג נ.סי.ד.

שיטת העבודה בעזרת המצגים היא קלה ונוחה למשתמש בה, וזו אחת הסיבות לכך, שהיחידה נפוצה כל כך בימינו. אדם, המקיש את הנתונים, רואה לפניו דף שלם, שניתן לבדיקה או לשינוי. בגמר ההקשה והבדיקה לוחצים על מקש ENTER, ורק אז נשלח תוכן המאגר מן המצג אל המחשב. אנו מניחים, כמובן, שבזיכרון המחשב מאוחסנת תכנית מתאימה, אשר "מבינה" את ה"הודעה" המגיעה אל הזיכרון. התכנית מסוגלת לעבד את הנתונים, שהגיעו אליה בתוך זמן קצר ביותר, אף בפחות משנייה אחת. תוצאת העיבוד נשלחת חזרה אל המצג ומוקרנת על המסך.

בעזרת המצג "משוחח" המשתמש עם המחשב - פעולה שלא יכולה היתה להתבצע עם אף לא אחת מיחידות הקלט/פלט, שנסקרו קודם. שיטת עבודה זו קרויה גם **עיבוד הידברותי** (INTERACTIVE PROCESSING). ללא מסופים, העיבוד נעשה **באצווה** (BATCH PROCESSING). הנתונים מגיעים בכמויות גדולות אל המחשב, והתכנית המאוחסנת מעבדת אותם ברצף.

המסופים מאפשרים **עיבוד מקוון** (ON-LINE), שבו ניתן לעבד רשומה בודדת, ולעדכן קובץ, אשר רשום בדיסק מגנטי, לפי תנועות בודדות. באצווה, נעשה עדכון קובץ פריטים במחסן בכמויות, פעם בשבוע או פעם ביום. כל התנועות של הוצאה או הכנסה של פריטים למחסן נאספות ונרשמות בכרטיסים, בתקליטון או בסרט מגנטי ומועברות לעדכון במרוכז. בתהליך זה ניתן לשלוף רשומה בודדת מקובץ הפריטים, אשר רשום בדיסק המגנטי, לעדכן אותה על המסך ולכתוב אותה חזרה למקומה בקובץ, מייד לאחר ביצוע הפעולה.

ישנם מסופים, שמותקן בהם **עט-אור** (LIGHT-PEN), אשר מכוון להקל על המשתמש בעבודתו. המשתמש בעט-אור "נוגע" בו במסך, במספר או בשורה הרצוים במקום להקיש את המספר או את השורה הנבחרים. היתרונות הם: קיצור זמן העבודה וצמצום השגיאות בקליטת הנתונים.

עיבוד מקוון - מהו?

נדגים עיבוד מקוון והידברותי על-ידי תיאור תהליך של הרשמת סטודנטים לאוניברסיטה. ביישום תהליך זה נשתמש בעט-אור, אך אפשר גם להשתמש במקלדת. הסטודנט מתבקש לשבת מול מסך, ולהקיש את המלה: **הרשמה**, ואח"כ ללחוץ על מקש ENTER. המחשב קולט את בקשתו ושולח אל המצג את המסך הבא:

הרשמת תלמידים

1 - תואר ראשון
2 - תואר שני
3 - תואר שלישי
4 - לימודי תעודה
5 - מסלולים מיוחדים
6 - לימודים שלא לקראת תואר

הנך מתבקש לגעת בעט בשורה הרצויה

נניח שהסטודנט נגע בעט אור בשורה מספר 1 (או הקיש את המספר 1) ואז יקבל "מסך" אחר:

מסי תעודת זהות :
שם :
כתובת :
מסי טלפון :
פקולטה מבוקשת :
1. מדעי הרוח
2. מדעי החברה
3. מדעי הטבע

עליך למלא את הפרטים האישיים ואח"כ לנגוע בעט בשורה של הפקולטה המבוקשת.

נניח ששם הסטודנט הוא עופר אברהמי ומסי תעודת הזהות שלו 11223344. לאחר שעופר מילא את פרטיו האישיים באמצעות המקלדת ואחר-כך נגע ב"מדעי החברה" או הקיש את המספר 2, התקבל המסך הבא, אשר ישא מעתה את שם הסטודנט ומסי תעודת הזהות שלו.

מס' 11223344 שם: עופר אברהמי
מדעי חברה: חוגי לימוד

1 - כלכלה　　　　　5 - אנטרופולוגייה
2 - סוציולוגייה
3 - סטטיסטיקה
4 - מנהל עסקים

הנך מתבקש לגעת בעט בחוג המבוקש.

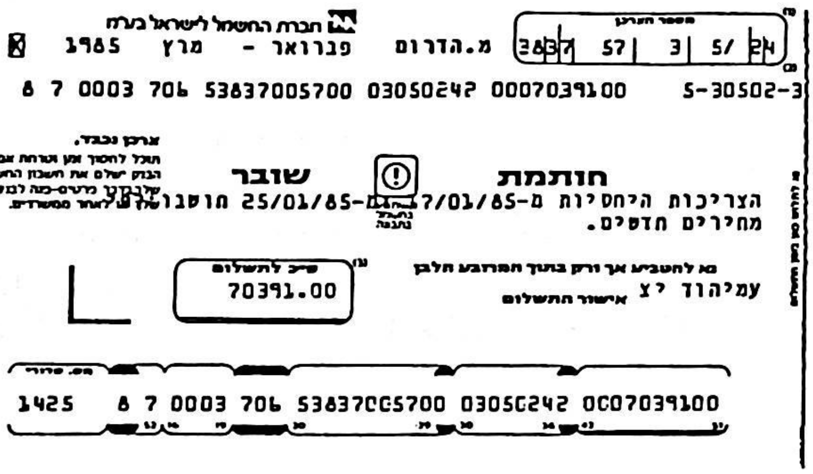
מעתה ימשך תהליך ההרשמה בצורת "דו-שיח" בין הסטודנט והמחשב. המחשב יפנה אל המועמד בצורה של הצגת רשימה של תשובות אפשריות, והמועמד יבחר בנגיעת עט-אור, או ע"י הקשת מספר התשובה את האפשרות הנראית לו. בתהליך זה מתקצר זמן ההרשמה עד כדי מחצית הזמן, שהיה דרוש בשיטות אחרות, ללא מחשב. במיוחד פחתו השגיאות, כי אין צורך להקיש מלים או משפטים אלא יש לבחור במלה או משפט הכתובים על המסך בצורה נכונה.

יתרונות דומים אפשר להשיג אם מקישים את מספר הפרט או השורה במסך. אפשר גם להשתמש בסמן במקום עט אור. מסיטים אותו אל המספר או אל השורה הרצויים ומקישים ENTER. המצג מעביר למחשב את המידע הזה לשם המשך התהליך, כפי שהיה לאחר נגיעה בעט אור.

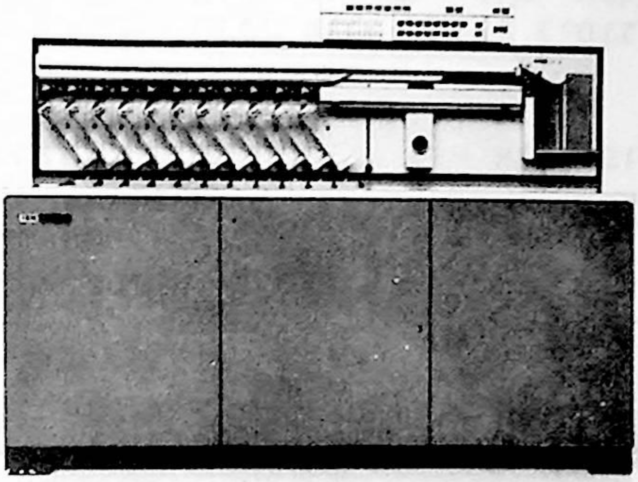
שימושים נוספים של יחידת המסך יוצגו בפרק 11, הדן בעיבוד נתונים בתקשורת.

4.7 קורא אופטי

<u>קורא אופטי</u> (OPTICAL CHARACTER READER - OCR) יכול לקרוא נתונים ממסמכים המודפסים במכונה, מסומנים, או כתובים ביד.



ציור 4-18 - טופס לקריאה אופטית



ציור 4-18.1 -
קורא אופטי

המסמך עובר ממחסנית הקליטה אל תחנת הסריקה האופטית, שבה מותקן מנגנון המורכב ממקור אור חזק, עדשות ומכשיר, אשר מזהה כל סימן על המסמך ומשווה אותו אל אוסף הסימנים המוכרים והמאוחסנים במערכת. הסימנים שזוהו מועברים אל זיכרון המחשב.

קיימים שלושה סוגי טפסים, שקורא אופטי יכול לקרוא:

- טופס המודפס על-ידי המחשב
  המשתמש אינו ממלא בו דבר, אלא דואג להחתמתו או לאישורו. לדוגמה, חשבון טלפון, שנשלח אל המנוי, ומוחזר לקורא האופטי, לאחר ששולם והוחתם בחותמת קופה.

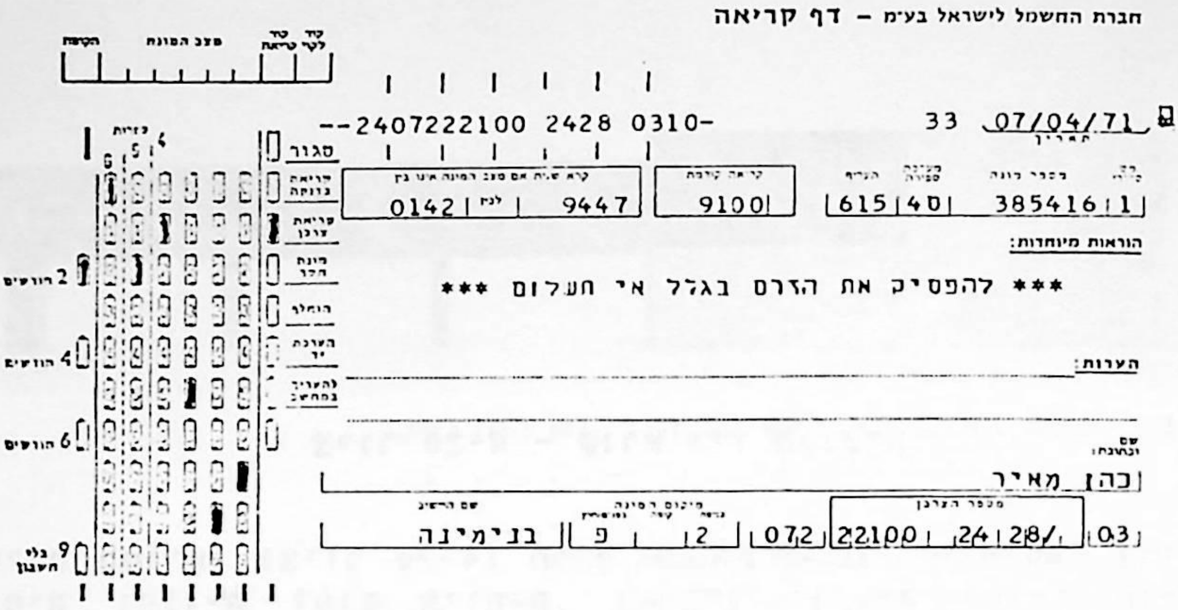
- טופס למילוי חלקי על-ידי המשתמש

  זהו טופס בעל תבנית קבועה ומוכרת למערכת, אשר המשתמש חייב להשלים בו פרטים. לעתים מתבטא מילוי הטופס בהעברה של קווי סימון. לדוגמה, טופס לקריאת מונה חשמל של חברת החשמל (ציור 4-19).

- טופס למילוי על-ידי המשתמש

  זהו טופס, שבו כל הפרטים ממולאים בכתב-יד או במכונת כתיבה. טפסים למילוי ביד היו נהוגים במשרדי ממשלה אחדים ובמחסני החלפים של חברת אגד.

הקוראים החדישים הינם משוכללים הרבה יותר מקודמיהם. הם כוללים גם יחידות בקרה ועיבוד, אשר מאפשרות ביצוע פעולות קלט, פלט וגם פעולות עיבוד בסיסיות.



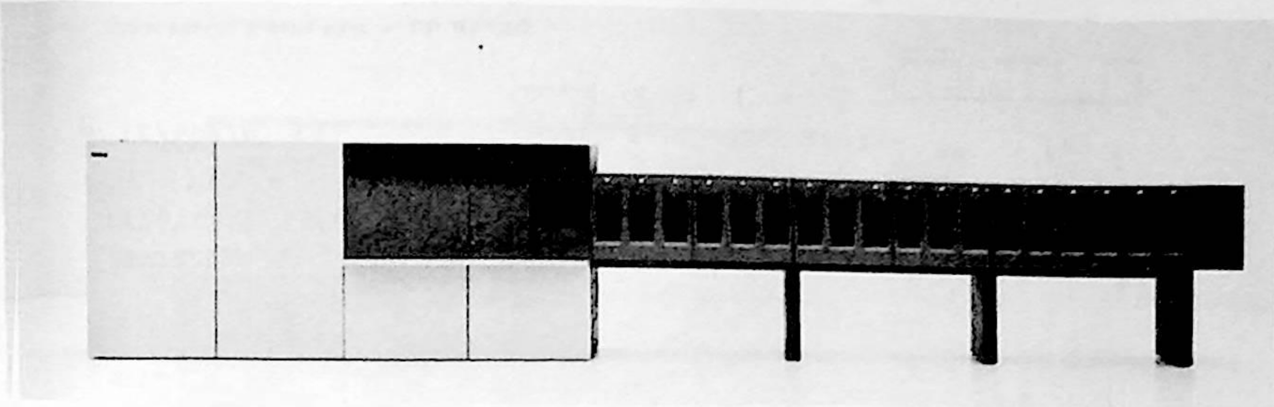
ציור 19-4 – טופס לרישום מצב המונה (חברת החשמל)

היתרון הבולט של יחידה זו על שאר יחידות הקלט/פלט הוא בכך, שכאן נחסכת תחנה בעיתית של קליטת הנתונים על ידי מערכת המחשב: העתקת נתוני המקור לכרטים או לתקליטון ע"י קלדנית. פעולה זו האריכה את זמן קליטת הנתונים ומטבע הדברים היתה מלווה בשגיאות אנוש. באמצעות הקורא האופטי "מצולם" טופס המקור כמות שהוא לתוך הזיכרון, ועל-ידי כך נחסך זמן יקר, שהוקדש להעתקה וקטן מספר הנתונים השגויים. אפשר לבטל לחלוטין את התיווך בהעברת נתונים למחשב בשיטה של עיבוד מקוון, שבו המשתמש, יוזם הנתונים, מזין אותם במסוף. דרך פעולה זו הוצגה בסעיף הקודם.

4.8 קורא דיו מגנטית

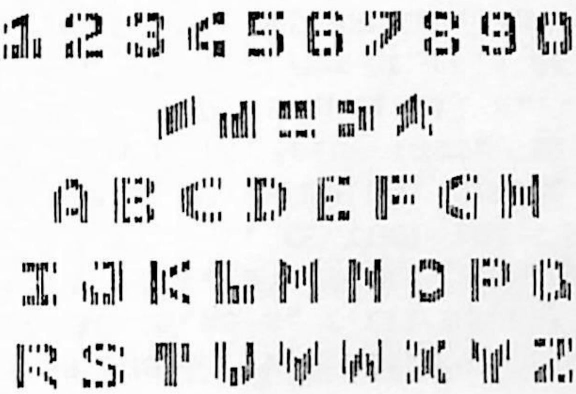
<u>קורא דיו מגנטית</u> (MAGNETIC INK CHARACTER READER - MICR) דומה לקורא האופטי.גם כאן נחסכת פעולת התיווך בין מסמך המקור לבין המחשב. מכונה זו יכולה לקרוא מסמכים, אשר הוכנו מראש לקריאה מגנטית. לשם כך מוטבעים הנתונים מראש בעזרת דיו מגנטית על הטופס או מוטבעים לאחר השימוש בטופס, אך לפני העברתו לקליטה.

אמינות ההטבעה מראש על הטפסים ומניעת אפשרות זיוף הביאו לכך שרוב הבנקים בחו"ל ובארץ משתמשים בשיטה של שיקים, המוטבעים בדיו-מגנטית . נזכור כמה מיליארדים של שיקים נכתבים מידי יום בכל העולם, ונבין את חשיבות הדבר בתחום זה של הפעילות העסקית.

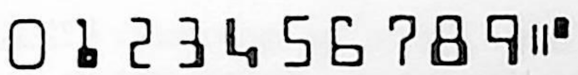


ציור 4-20 - קורא דיו מגנטית

היחידה הקוראת (ציור 4-20) מזהה את הסימנים, המוטבעים בדיו מגנטית בתבנית דפוס מיוחדת. בארה"ב ובקנדה נפוצה שיטת הסימנים E13B, ואילו בישראל נפוצה השיטה האירופאית, המכונה CMC7.



CMC7



E13B

ציור 4-21 - תבניות שונות של תווים לקריאה מגנטית

יחידת ההזנה מזינה את המסמכים למכונה במהירות של 1,500 עד 2,000 מסמכים בדקה. יחידת העברה וקריאה קוראת את הרישום המגנטי, ובאמצעות מנגנון נוסף מאפשרת למיין את המסמכים לקבוצות שונות. מונה המסמכים מציין את מספר המסמכים שנקראו.

הקורא האופטי יכול לפעול במופרד מן המחשב (OFF-LINE) ואז הוא משמש למיון ולרישום על סרט מגנטי. כאשר הוא מחובר למחשב (ON-LINE), הוא מאפשר, תחת פיקוח התכנית שבזיכרון המחשב, לבצע בדיקות לוגיות, להפריד מסמכים שגויים ולהפיק דו"ח על המסמכים השגויים.

<u>פרק 4 - שאלות</u>

1. מנה את החסרונות של הכרטים המנוקב.

2. תאר שימוש בסרטים מגנטיים.

3. לפי מה מבדילים בין סוגי המדפסות השונים?

4. קובץ התושבים בעיר מסוימת מכיל 400,000 רשומות. בכל רשומה כ-250 תווים. הקובץ מאוחסן בסרטים מגנטיים. בכמה זמן ייקרא הקובץ לזיכרון המחשב, ובמשך כמה זמן יודפס הקובץ במדפסת שרשרת? בחר מהירויות עיבוד כרצונך לפי הנתונים בגוף החומר.

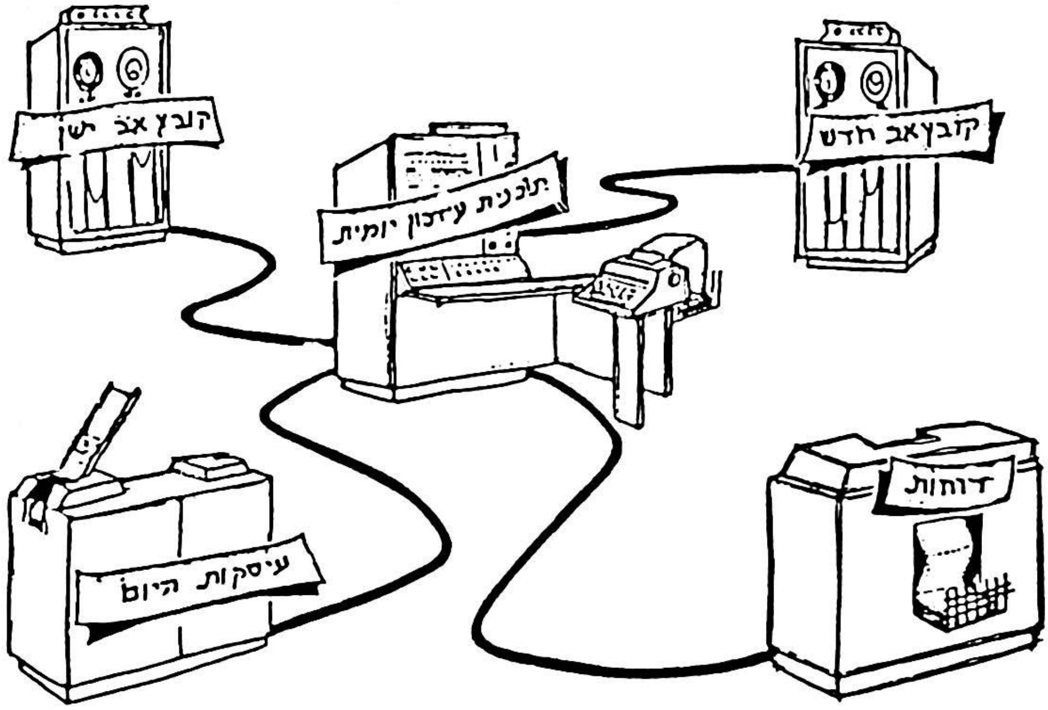
5. באיזה מקרה תעדיף תקליטון על פני סרט מגנטי?

6. מדוע נקרא הדיסק המגנטי גם בשם "זיכרון עזר"?

7. באיזה מקרה אי-אפשר לוותר על השימוש בדיסקים מגנטיים?

8. ציין את ההבדלים בין עיבוד מקוון לבין עיבוד באצווה.

9. "הקורא האופטי הוא מכשיר יקר ואיטי מאד. מהירות הקריאה שלו אינה עולה על 2,000 תווים בשנייה. אינני מבין, מדוע משתמשים בו עדיין!"
הגב על מובאה זו.


קובץ אב ישן
תוכנית עדכון יומית
קובץ אב חדש
עיסקות היום
דוחות

פרק 5 – עקרונות התכנון והתכנות

בשני הפרקים הקודמים ניתנה סקירה מקיפה על מערכת מחשב מבחינת החמרה (HARDWARE), כלומר המכונות והציוד, בהם משתמשים ביחידת עיבוד נתונים. אך כבר בפרק הראשון של ספר זה הודגש, כי אין כל תועלת במערכת המחשב (תהיה גדולה ומקיפה, ככל שתהיה), אם לא הוזנה לזיכרון תכנית (PROGRAM), אשר תעבד את נתוני הקלט ותפיק מהם את המידע הדרוש.

מי כותב את התכניות? כיצד כותבים אותן? האם אפשר לכתוב אין סוף תכניות?

בפרקים הבאים נעסוק בכל הקשור בתכנה. במונח תכנה (SOFTWARE) מתכוונים לכל התכניות, הנכתבות על-פי-רב ע"י תכניתנים. הן כוללות הוראות שונות, שבאות לפי סדר הגיוני מסויים, ומתבצעות על ידי המחשב בסדר זה. בנוסף, נדון בכל התהליכים הקשורים בתכנון ובפיתוח התכנה. קיימת קבוצה נוספת של תכניות, שמכוונות לספק שירותים שונים, כדי לייעל את עבודת המערכת. על תכניות אלו יורחב הדיון בפרק 8.

התהליכים, שקשורים בתכנון ובפיתוח התכנה, כוללים מספר שלבים. הם מתחילים בשלב של הצגת הצורך למיכון תהליכי עיבוד נתונים, ומסתיימים בשלב של הפעלת המערכת כולה לפי דרישות הצרכן, שהזמין אותה.

בפרק זה יוסברו עקרונות התכנון, ויוצגו באופן כללי השלבים השונים בהכנת תכניות מחשב.

5.1 עקרונות התכנון

רבים מאתנו היו רשומים בילדותם בספרייה של בית-הספר, בו למדו, או בספרייה הציבורית שבמקום מגוריהם. אחדים מאתנו עדיין נוהגים לשאול ספרי קריאה וספרי עיון בספרייה הסמוכה לביתם. באחד מביקורי בספרייה העירונית בעיר מגורי, סיפרה לי הספרנית, שמתכוונים "למכן" את הספרייה. היא ביקשה לדעת, כיצד יבוא לידי ביטוי מיכון הספרייה. הסברתי לה, שאוכל להשיב על שאלתה לאחר בירור של פרטים אחדים, וביקשתי לדעת, כיצד מתנהל רישום הספרים.

הספרנית הסבירה לי, שקיימת בספרייה כרטסת, ובה כרטיס לכל ספר. בראש הכרטיס רשומים שם המחבר, שם הספר ומספר הספר

(שנקרא "מספר קטלוגי"). בחלקו התחתון של הכרטיס רשומים שמות השואלים, תאריך ההשאלה של הספר ותאריך החזרתו.

המספור של הספרים במספרים קטלוגיים דרוש, כדי שאפשר יהיה להזמין ספרים ולעקוב אחר השאלתם. הקטלוג משרת את הקוראים, שאינם יודעים מראש איזה ספר היו רוצים לקרוא. הם מעיינים בו ומבקשים מהספרנית את הספר המבוקש על ידם.

מצויד בפרטים, שקיבלתי בספרייה, פניתי למשרדי העירייה, כדי לבדוק כיצד משתמשים במחשב, שעומד לרשותה.

בבית העירייה הוסבר לי, שהשימוש במחשב קיים זה שנים אחדות. המחשב שימש עד אותו זמן לגביית מסי העירייה, למעקב אחר ניצול סעיפי התקציב ולעבודות שונות במחלקת הנהלת חשבונות. השנה הוחלט "למכן" או "למחשב" גם את הספרייה.

הבה נעצור לרגע. מי קובע בגוף או מוסד כמו העירייה, מה למכן, מתי למכן, כיצד למכן. מה פירוש הביטוי "למכן"?

העירייה הינה גוף גדול יחסית. כל מוסד, ארגון או גוף קיים כדי להשיג מטרה מסויימת. כל גוף זקוק למשאבים כדי לבצע את עבודתו. העירייה למשל, זקוקה לכסף כדי להעניק שירותים שונים לאוכלוסייה: שירותי נקיון, חינוך, תאורה ברחובות ועוד. העירייה מפעילה מוסדות ואנשים, שעובדים בתפקידים שונים ומופעלים במחלקות השונות כגון מחי החינוך, מחי מים ומאור ועוד. הנהלת העירייה מוציאה הוראות שונות למנהלי המחלקות בעניין תפעול ותפקוד אותה מחלקה. מנהל המחלקה מפעיל את האנשים. האנשים חייבים לדווח על ביצוע העבודה ועל תוצאותיה. מידע זה מועבר דרך מנהלי המחלקות אל הנהלת העירייה. המידע חייב לזרום מלמעלה למטה ומלמטה למעלה, כלומר ממנהלי המחלקות לאנשי הביצוע, ומאנשי הביצוע אל דרגי הניהול.

חלוקת המשאבים בצורה נכונה, הזרמת המידע מהדרג התפעולי אל הדרגים הניהוליים ולהיפך, הצורך לשקול את השיקולים הנכונים ולקבל את ההחלטות הנכונות הובילו את מנהלי הארגונים השונים להפעלת כלים, אשר יעזרו להם בביצוע תפקידיהם ובהשגת משימותיהם.

אחד מהכלים החשובים, היכולים לסייע לארגון בביצוע משימותיו ובפתרון בעיותיו, הוא המחשב. מערכת המחשב והיחידה הארגונית האחראית לתפעול מערכת זו - <u>היחידה לעיבוד נתונים</u> - נחשבות היום לאחד מספקי השירות החשובים ביותר בכל ארגון וארגון. ביחידה זו אפשר למצוא חמש קבוצות של ממלאי תפקידים:

- מנהל יחידת המחשב.
- מנתח מערכות (SYSTEMS ANALYST)

- תכניתן (PROGRAMMER)

- מפעיל (OPERATOR)

- פקיד ביצוע

<u>מנהל היחידה</u> אחראי כלפי הנהלת הארגון על התפעול היעיל של היחידה כולה.

<u>מנתח מערכות</u> הינו איש הקשר המקצועי בין יחידת המחשב לבין ציבור המשתמשים והצרכנים באותו ארגון. דרישות ובעיות הצרכנים מתוארות במדויק בתיקי תכנון. תיקים אלה נמסרים לתכניתנים.

<u>התכניתן</u> כותב תכניות למחשב בהתאם להנחיות שבתיקי התכנון.

<u>המפעיל</u> אחראי להפעלה התקינה של הציוד (HARDWARE) הכולל יחידות קלט/פלט, יע"מ, יחידות בקרה ויחידות תקשורת למיניהן. תכניות המחשב מתבצעות במערכת זו.

<u>פקיד הביצוע</u> אחראי לאיסוף הנתונים, הזנתם אל המחשב והפצת הפלט של המחשב אל המשתמשים במחלקות השונות של הארגון.

עתה נחזור לספרייה העירונית. הנהלת העירייה קיבלה דיווחים על עבודתה היעילה של יחידת המחשב. כן נתברר, שרוב התכניות שהיו קשורות במחי הגבייה, מחי הנה"ח ומחי שכר - פועלות כהלכה, ועכשיו ניתן להרחיב את מעגל "המחשוב" בעירייה. הנהלת העירייה בחרה, בין השאר, גם בספרייה העירונית כיחידה, שיש למכן אותה.

כדי למכן מערכת כמו הספרייה העירונית, יש לבצע <u>ניתוח מערכת</u> (SYSTEM ANALYSIS). ניתוח מערכת הינו תהליך ארוך המורכב בדרך כלל מחמישה שלבים:

1. <u>הגדרת המשימה</u>

כל תהליך "מחשוב" או "מיכון" מתחיל <u>ביזמה.</u> במקרה שלפנינו הנהלת העירייה היתה היוזמת למחשוב הספרייה, אולם אין די ביזמה. ההנהלה חייבת לקבוע את היעדים המדויקים של תהליך המחשוב. האם למכן את הזמנות הספרייה החדשות או רק את השאלת הספרים? האם יחידת המחשב תספק דו"חות מצב לספרייה כל יום, פעם בשבוע או פעם בחודש? מהן המגבלות התקציביות לכל התהליך? האם יש צורך לקבוע תאריך סיום לפעולות הללו? כל ההחלטות בכל הסוגיות הללו חייבות להיות ברורות וחד-משמעיות. מנתח המערכות מייעץ בשלב זה להנהלת העירייה, מספק מידע על העלויות, מעריך את משך הביצוע של הפרויקט, אך ההחלטה והאישור הסופי באים מטעם ההנהלה.

2. <u>לימוד המצב הקיים</u>

כאשר מנתח מערכות לומד את המצב הקיים, הוא חייב בראש ובראשונה לאבחן את <u>הבעיות</u> ואת הגורמים לאותן בעיות. עלינו לזכור, שברוב המקרים היזמה למיכון מתעוררת, כאשר המצב הקיים אינו משביע רצון, והאחראים בארגון מתכוונים לשנות את המצב הקיים. מנתח המערכות מתבקש לבחון גם את התהליכים הקיימים באותו גוף או באותה תת-מערכת שמנתחים. כמו כן נבדק המבנה, הנוהלים והצינורות בהם זורם המידע.

השאלות, אשר הופנו אל הספרנית, יכולות לשמש דוגמה לשיטה בה נאספים נתונים על המצב הקיים. מנתח המערכות בעירייה ילמד את הדרכים בהן פועלת הספריה, ימצא מהי השיטה להזמנת ספרים, מהי צורת הרישום בקטלוג, ועוד. בתוך כדי הלימוד של הקיים אפשר למצוא גם את הבעיות. ייתכן שהקטלוג לוקה בחסר ויש להרחיב אותו, ייתכן שכרטסת אחת איננה מספיקה וצריך ליצור שתי כרטסות נפרדות, כרטסת לפי שמות השואלים וכרטסת לפי המספרים הקטלוגיים. במקרה כזה, השינוי בכרטסת או במבנה הקטלוג ייקבעו בתוך כדי התייעצות עם הספרנית, האחראית להפעלתה התקינה של הספריה.

3. <u>חקר ישימות</u>

הנחת היסוד של מנתח המערכות היא, שלכל בעיה אפשר למצוא פתרון או אפילו כמה פתרונות. בשלב זה הוא מתקדם צעד אחר צעד. הצעד הראשון מכוון למציאת הדרכים והכלים בהם אפשר לפתור את הבעיות, שאותרו קודם לכן. בצעד הבא הוא בוחן, באיזו מידה תואמים הפתרונות את הדרישות, שהוגדרו בראשית התהליך.
תהליך זה קרוי <u>חקר ישימות</u> (FEASIBILITY STUDY).

4. <u>תכנון</u>

בשלב זה נקבעים המשאבים הדרושים לביצוע המשימה, שלבי העבודה שיתבצעו בעתיד, לפי לוח זמנים. חלק זה של העבודה חייב להתבצע תוך שיתוף פעולה הדוק עם הצרכנים. במקרה שלנו יש לתכנן את הפעילויות יחד עם הספרנית, אשר תעבוד בעתיד לפי השיטה המוצעת.

זה גם הזמן המתאים לקבל את אישור ההנהלה לתכנית. הסיבה ברורה: ייתכן שמחשוב הספרייה כרוך בהוצאות כספיות כאלו, שאין העירייה מסוגלת לעמוד בהן. במקרה כזה אולי כדאי לחפש מקורות מימון נוספים, או לחלופין להכניס שינויים בתכנית המוצעת.

5. <u>הכנת המפרטים</u>

בשלב האחרון של ניתוח המערכת נוצרים תיקי המפרטים, אשר

כוללים תיאור מדויק של הקלט, הפלט והוראות לביצוע התכנות. התיקים נמסרים לתכניתנים, אשר יכתבו תכניות מחשב לפי הוראות אלו.

במקרה של הספרייה העירונית חייב מנתח המערכות לתאר במדויק את מבני הקבצים. הוא חייב לתת תשובות מדוייקות לשאלות הבאות: מה ייכתב בקטלוג ולפי איזה סדר? האם יופיעו רק שמות הספרים ומחבריהם, או יצוין גם שם בית ההוצאה? ובאיזו שנה יצא הספר לאור? מה תהיה צורת הדו"ח אותו תקבל הספרנית?

בתיקי המפרט תיכתבנה הוראות מדויקות לגבי כתיבת התכנית. תוגדר הדרך, בה יטפל המחשב בהזמנות ספרים, ייקבע הסעיף התקציבי אשר יחויב בכל מקרה, ומי יזוכה בתשלום ועוד ועוד.

5.2 תרשימי עבודה

תכניות המחשב מורכבות מסדרה ארוכה של הוראות, אשר ערוכות בסדר הגיוני מסוים. סדר זה נקבע על ידי המתכנן/מנתח המערכות. כדי למנוע אי-הבנות או פירושים בלתי נכונים נוהגים לצרף לתיק התכנון מעין "ראשי פרקים לעבודה" בצורת תרשים. תרשים כזה נקרא תרשים עבודה.

מבחינים בין שני סוגי תרשימים:

1. תרשים זרימה (SYSTEM FLOWCHART)

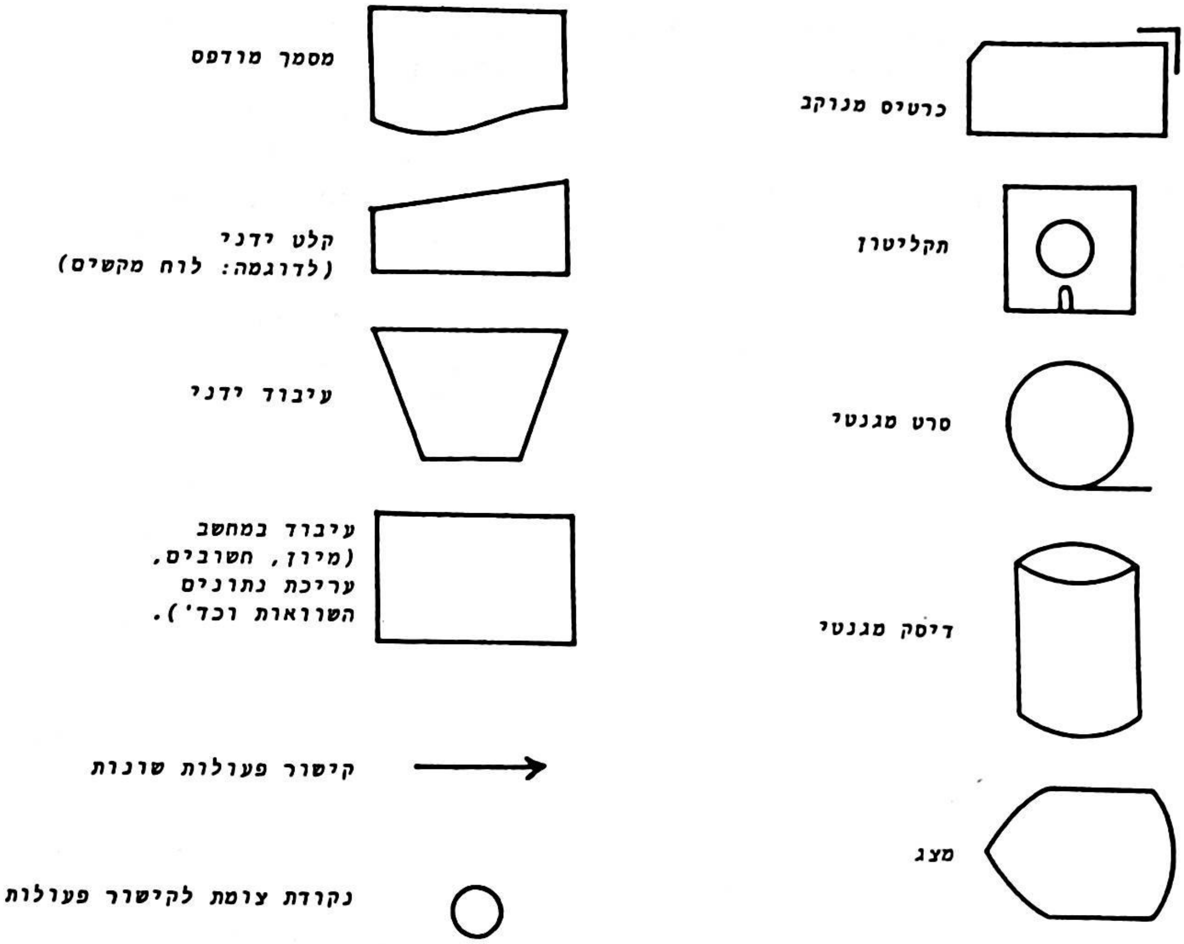
בתרשים זה מתוארים סוגי הקבצים השונים, המשתתפים בעיבוד וזרימת הנתונים מקבצי הקלט, דרך היע"מ אל קבצי הפלט. בתרשים כזה משתמשים ברוב הארצות והארגונים בסימנים מקובלים למטרה זו. הסימנים, המבוססים על צורות הנדסיות פשוטות, מוצגים בציור 1-5.

2. תרשים תכנית (PROGRAM FLOWCHART)

תרשים זה נעשה ע"י התכניתן בשלב מאוחר יותר. לכן יוסברו סימני התרשים הזה והשימוש בהם בסעיף הבא.

תרשים הזרימה הינו חלק חשוב ביותר של תיק התכנון. יתרונו בכך שהוא מציג את הנתונים ותהליכי העיבוד שלהם בצורה מתומצתת, אך יחד עם זאת - ברורה מאד.

הבה נראה קטע מתוך תיק התכנון, אשר מציג שימוש בתרשים זרימה, בקשר למיכון הספרייה:



ציור 5-1 - סימנים מקובלים בתרשימי זרימה

<u>ספרים חדשים</u>

א. <u>קלט</u>

אל ספריית העירייה מגיעים כל הספרים החדשים כאשר אל כל אחד מהם צמוד כרטיס מנוקב. הכרטיס מכיל שבעה שדות (ראה ציור 5-2). שדה הקוד מתייחס למדורי הספרייה, לפי הטבלה הבאה:

קוד 1 - שירה

קוד 2 - ספרות יפה

קוד 3 - ספרי ילדים

קוד 4 - ספרי מדע

קוד 5 - ספרות בדיונית

קוד	שם הספר	שם המחבר	שנת הוצאה	שם מו"ל	מחיר	תאריך
3	אב 1	מחבר 1	1984	מול 1	5780	1.8.84
2	אב 2	מחבר 4	1983	מול 1	3270	1.6.84

ציור 2-5 - מבנה רשומת ספר

ב: <u>עיבוד</u>

יש לקרוא את הכרטיסים ולהדפיס דו"ח שבועי לפי התיאור שלהלן. בנוסף לכך יש להעלות את רשומת הספר על סרט מגנטי לצורך העיבוד החודשי (חיוב תקציבי, זיכוי ספקים וכוי). המספר הקטלוגי ינתן ע"י הספרנית.

ג. <u>פלט</u>

מבנה הדו"ח השבועי

תאריך ..... ספרייה עירונית מס' ..... דו"ח שבועי

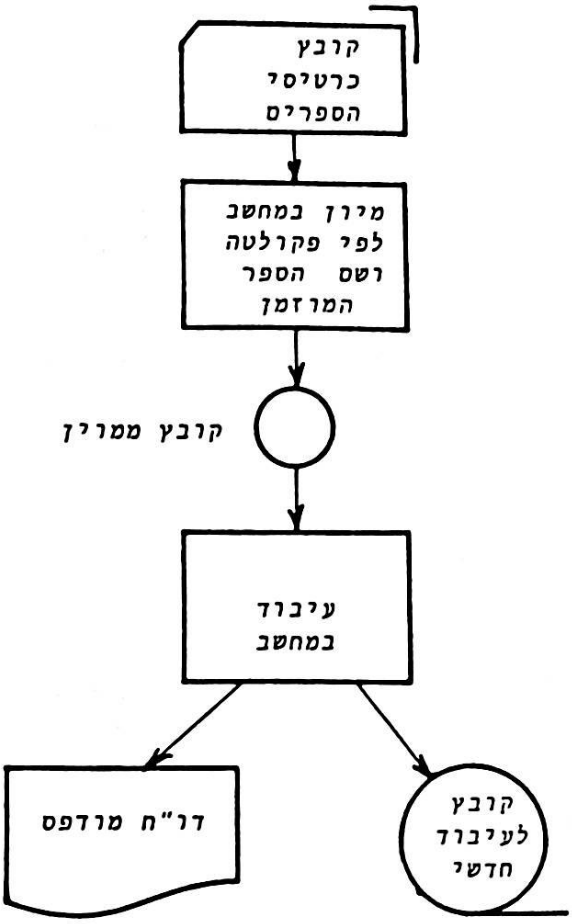
מדור	תאריך הזמנה	שם המחבר	שם הספר
מדע	1.7.84	מחבר 3	אב 4

ציור 3-5 - מבנה דו"ח

בסוף הדו"ח יש להדפיס את סה"כ הספרים, שהגיעו לספרייה, לפי מדורים. אם חסר קוד בכרטיס, יש להדפיס:

"** חסר קוד מדור **"

ד. <u>תרשים זרימה</u>



ציור 4-5 - תרשים זרימה

5.3 תהליך התכנות

לאחר שמנתח המערכות מכין את תיקי התכנות ומוסר אותם לתכניתן, מתחיל תהליך <u>התכנות</u> (PROGRAMMING). תכנות הינו מושג כולל להעברת הדרישות לעיבוד נתונים אל תכנית עבודה, אשר תוזן לזיכרון המחשב. <u>התכנית</u> (PROGRAM) מוגדרת כסדרת הוראות, הגורמות למערכת מחשב לבצע פעולות מסוימות, בסדר הגיוני מסוים, שתכליתן להגיע לתוצאה המוגדרת, או לחילופין, לפתרון בעייה מוגדרת.

תהליך התכנות כולל שמונה שלבים:

1. תכנון הפתרון.

2. הכנת תרשימים.

3. כתיבת תכנית בשפת תכנות כלשהי.

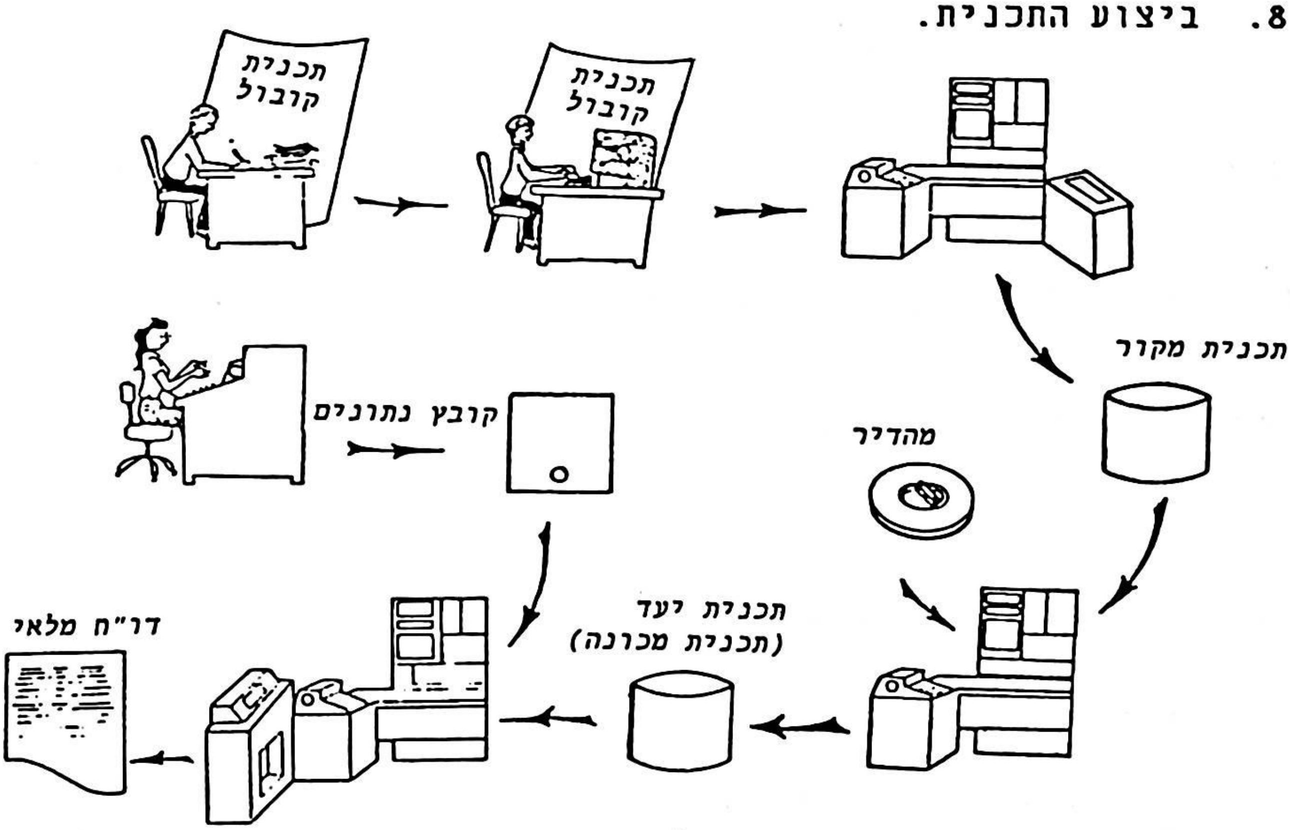
4. הזנת התכנית לזיכרון המחשב.

5. תרגום התכנית לשפת מחשב (הידור).

6. הרצת ניסוי.

7. ניפוי שגיאות.

8. ביצוע התכנית.



ציור 5-5 - שלבים בהכנת תכנית

הקורא ימצא את התיאור של שמונת השלבים בפרק זה. הסבר נרחב יותר על תרשימים נמצא בפרק 6, ועל שפות תכנות - בפרק 7.

<u>שלב 1 - תכנון הפתרון</u>

מלאכת התכנות, בין אם היא נעשית בידי אישה או גבר, כרוכה בראש ובראשונה בחשיבה רבה. לא במקרה בחרה חברת יבמ כסיסמתה את המלה: <u>חשוב</u> - THINK.

הצעד הראשון במלאכה זו - הבנת הבעיה וקביעת התהליכים, שיביאו לפתרונה. כאן יש לקחת בחשבון באיזו מערכת מחשב תופעל ("תרוץ") התכנית, מהן הפעולות השונות, שניתנות לביצוע באותה מערכת, ובאילו כלים ישתמש התכניתן לפיתוח התכנה: סוגי תרשימים, שפת תכנות, טבלאות עזר שונות וכו'.

בשלב זה יש להתחשב בשבע נקודות חשובות:

1. <u>הקצאת זיכרון</u> לקליטת התכנית, הנתונים ושטחי העבודה השונים.

2. טיפול בקובצי הקלט, הכולל בדיקה ראשונה של הקבצים, ולעיתים הסבת הנתונים לצורה אחרת, הנוחה לטיפול בתכנית.

3. קביעה של נתוני עזר כגון טבלאות, קובצי ביניים וקבועים שונים (CONSTANTS) לכותרות, הדפסות ועוד.

4. קביעת הדרכים לטיפול במקרה של תקלות חיצוניות, בלתי צפויות או תקלות בעת ביצוע התכנית.

5. קביעה של תבנית הפלט, אשר נשמר במערכת כקלט לעיבודים מאוחרים יותר.

6. קביעה של תבנית הפלט לשימוש בני-אדם, כמו דו"חות ותדפיסים שונים.

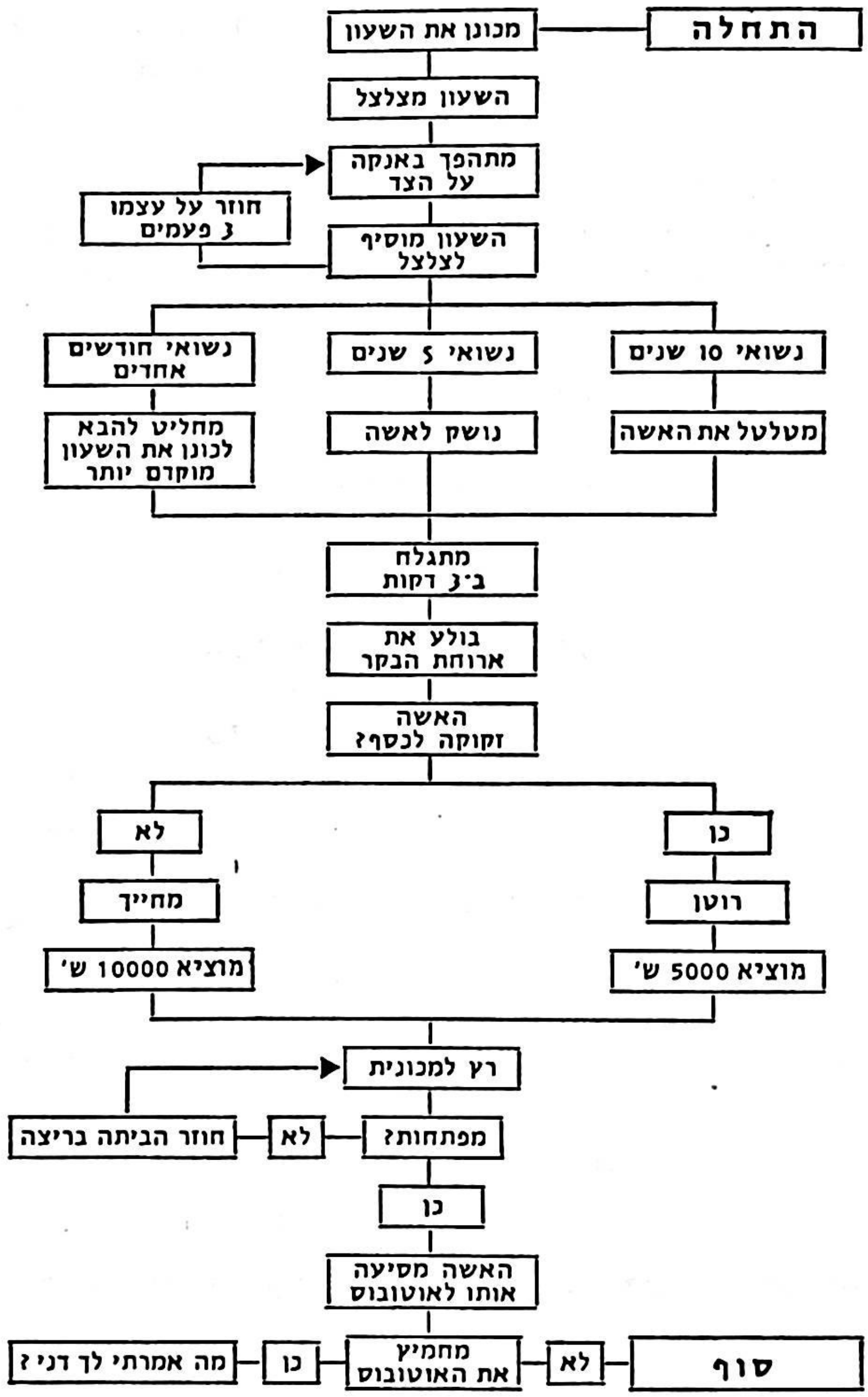
7. קביעת הדרישות לאימות החומר ודרגת הדיוק, הדרושה בחישובים.

בשלב זה התכניתן אינו "כותב" עדיין את התכנית. הוא אוסף מידע מכל מקום אפשרי, ומתכונן לקראת השלבים הבאים של עבודתו. ארגון החומר והתאמתו לדרישות, שהוצגו בתיקי התכנות בתוך כדי מיון קפדני - הן הפעולות, שמאפיינות שלב ראשון זה. נדרשים בו תכנון והכנה, ולמרות הזמן והמאמצים הרבים, המושקעים כאן, עדיין לא נכתבה אפילו שורה אחת מן התכנית. בסיום השלב מתגבשת שיטה לכתיבת התכנית, ואז ניגש התכניתן לכתיבה.

שלב 2 - הכנת התרשים

תכנית מחשב, המורכבת מסדרה של הוראות, הינה בדרך כלל ארוכה ומסובכת. לעיתים קרובות המעבר מפעולה מסוימת אל פעולה אחרת אינו ישיר או קבוע, ולכן גם מספר ההסתעפויות בתכנית הוא רב. על כותב התכנית לתאר את עבודתו ב"ראשי פרקים", כי התיאור המילולי לבדו, אשר תלוי בכושר ההתבטאות של הכותב, יכול להיות מסורבל ולא ברור די צרכו. ראשי פרקים אלה יירשמו בצורות הנדסיות שונות (מלבן, מעוין, מקבילית), אשר יחוברו על ידי חצים.

התרשים משמש גם לתיאור אלגוריתם (ALGORITHM). אלגוריתם מוגדר כסדרת הוראות, המציינות תהליכים לוגיים לפתרון בעיה כלשהי. להלן דוגמה לאלגוריתם, המתאר פתרון לבעיה יומיומית של חלק גדול מהאנשות - כיצד לצאת לעבודה בבוקר.



ציור 5-6 - יציאה לעבודה בבוקר

תכונות האלגוריתם:

- כל הוראה חייבת להיות חד-משמעית. למשל, אין לתת הוראה כגון: "הבא מעט מים", או "הוסף עגבניות רבות לסלט".

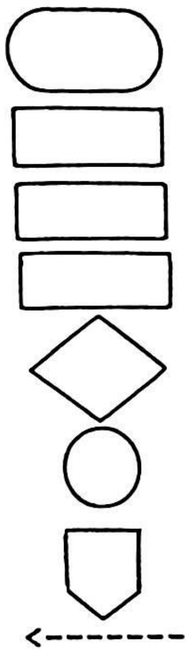
- סדר הביצוע של ההוראות חייב להיות מוגדר באופן חד-משמעי. בדוגמה לעיל, הפעולה "צא למכונית" חייבת להקדים את הפעולה "סע לעבודה".

- יש למצות באלגוריתם את <u>כל</u> האפשרויות ההגיוניות, הקשורות בביצוע התהליך. בציור 5-6, למשל, תנאי זה אינו מתקיים. הקורא מתבקש למצוא מהי השגיאה.

• ביצוע התהליך חייב להיות מורכב ממספר סופי של צעדים. למשל אם האישה אינה מתעוררת, אי-אפשר להמשיך את ביצוע התכנית.

בציור 5-6 בולט לעין, שרוב ההוראות נכתבו בתוך מלבן. בגלל סיבה זו נוהגים לקרוא לתרשים מסוג זה גם **תרשים מלבני** (BLOCK DIAGRAM). למרות שפרק 6 כולו יוקדש לתרשימים, נביא כאן את הסימנים המקובלים ביצירת תרשימים מלבניים, שהם רק אחת הדרכים להצגה של הפתרון. דוגמה לשימוש בתרשים כזה תמחיש את הקשר בין שלבים מספר 1 ומספר 3 בהכנת תכנית מחשב.

הסימנים, המקובלים לתרשים המלבני, הם:



התחלה/סוף

הגדרות, פעולות הכנה

עיבוד

פעולות קלט, פלט

שאלה או נקודת החלטה

נקודת חיבור

מעבר אל/מ-דף אחר

חץ המקשר בין הצורות

בדוגמה שלהלן יוצג בתחילה תיאור הבעיה, אח"כ תרשים זרימה שלה, תרשים מלבני, ולבסוף הסבר לתרשים המלבני.

בעיה : הדפסת רשומות מקובץ

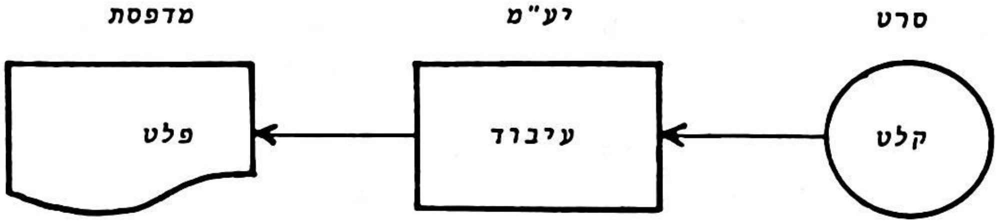
1. **תיאור הבעיה**

א. **נתון:** קובץ סטודנטים בסרט מגנטי. לכל סטודנט יש רשומה.

ב. **פלט:** דו"ח, המכיל את רשימת כל הסטודנטים, כל אחד בשורה נפרדת.

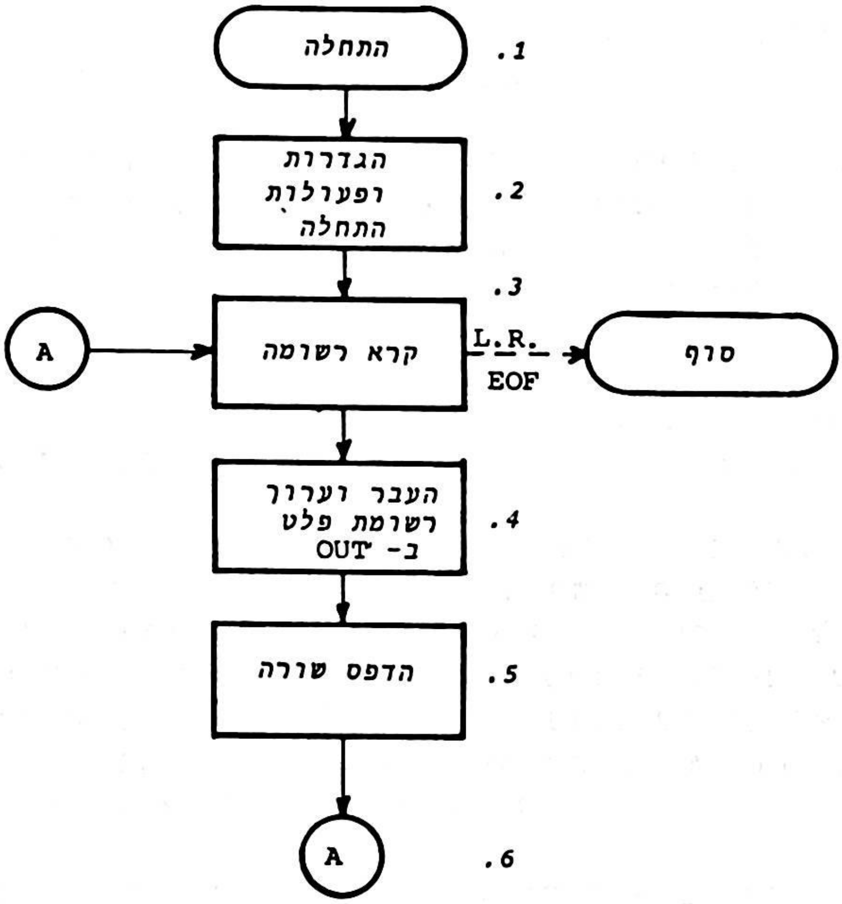
ג. **עיבוד:** הדפסה. יש לעצור בסיום פעולת ההדפסה.

2. <u>תרשים הזרימה מתאר את תהליך הפתרון</u>



ציור 5-7 - תרשים זרימה כללי

3. <u>התרשים המלבני מפרט את שלבי העיבוד</u>



ציור 5-8 - תרשים מלבני

<u>ניתוח הפתרון</u>
(המספרים מציינים את תבניות התרשים שבציור 5-8)

1. <u>התחלה</u> - כל תרשים מלבני מתחיל בסימן זה. <u>חץ</u> מקשר אותו אל השלב הבא.

2. הגדרות - בהגדרות נוהגים לציין שני סוגי הגדרות: הסוג הראשון מתייחס לסביבת העיבוד (ENVIRONMENT). כאן יש לציין, אילו יחידות קלט ופלט תשותפנה בעיבוד זה (היות שכלל זה אינו ישים לגבי כל שפות התכנות - לא ייכתב דבר במלבן, פרט למלה "הגדרות").

הסוג השני מתייחס אל השדות בזיכרון המחשב, שנדרשים בעיבוד ספציפי זה. בדוגמה שלפנינו הראינו שני שדות כאלו בשם IN ו-OUT. בקבוצת פעולות זו כלולות גם פעולות התחלה (INITIALIZATION) שונות. פעולות ההתחלה המקובלות הן איפוס מונים או הקניית ערכים התחלתיים למשתנים, הדפסת כותרות ופעולות שונות אחרות הנדרשות לפי תנאי הבעיה. בדרך כלל פעולות אלו מתבצעות פעם אחת בלבד בתחילת התכנית, ואין חוזרים עליהן.

3. קרא רשומה - זוהי פעולת קלט, אשר גורמת להעברת הנתונים מכונן הסרטים אל הזיכרון הפנימי במחשב. עם ביצוע פעולה זו מועבר תוכן הרשומה הראשונה לשדה בשם IN.

4. העבר וערוך רשומה בשטח OUT - העברה של הנתונים לשדה בשם OUT. פעולה זו כוללת עריכת השורה בתבנית השורה המודפסת, והיא הכרחית במקרה שלפנינו. הנתונים נכתבו על הסרט ללא רווחים, בצורה צפופה, ואם היינו מדפיסים אותם, כפי שהם מצויים בסרט, ספק אם יכולנו להבין את הכתוב.

5. הדפס שורה - זוהי פעולת פלט, הגורמת להעברת הנתונים אל המדפסת, קידום הנייר לשורה הבאה והדפסת השורה על הנייר.

6. סיעוף לנקודה A - בשלב זה של העיבוד ביצענו את כל הפעולות הדרושות לקריאת רשומה אחת מן הסרט והדפסתה על נייר המדפסת. כדי להדפיס את הרשומות הבאות עלינו לחזור על הפעולות הללו, וליתר דיוק על פעולות 3 עד 5. נכנה את פעולת הקריאה כפעולה מס' A, ואזי נוכל למשוך חץ מ"הדפס שורה" אל עיגול, ובתוכו נרשום את הסימן A, שהוא נקודת חיבור עם העיגול A, ליד פעולת הקריאה.

מעתה תחזור המערכת על ביצוע הפעולות 5,4,3 בהתאם לחצים, המקשרים את הפעולות הללו ביניהן: תקרא רשומה, תערוך אותה כפלט, תדפיס אותה על המדפסת ותחזור חלילה.

עד מתי נהיה מעוניינים בביצוע המחזור של פעולות 5,4,3 ? כמובן, עד סוף הקובץ, עד שנשלים את הפעולות, המתייחסות לרשומה אחרונה שבקובץ.

על הסרט המגנטי נמצא סימן מיוחד, המציין את סוף הקובץ (END OF FILE), והמערכת תבחין בכך בתוך כדי קריאת הסרט.

יש לציין ליד פעולת הקריאה את הפעולות, הדרושות לביצוע, במצב של זיהוי סוף הקובץ. במקרה שלנו - לעצור.

7. סוף - פעולה זו נמצאת מימין להוראת הקריאה, ומחוברת באמצעות חץ מקווקו אל התחנה האחרונה. לעיתים נבדק סוף הקובץ במפורש ע"י שאלה. מעל החץ נוהגים לציין סוף קובץ - EOF (END OF FILE) או רשומה אחרונה (LAST RECORD).

שלב 3 - כתיבת התכנית בשפת תכנות (שלב 1 בציור 5-5)

לאחר הכנת התרשים, התכניתן כותב את התכנית באחת משפות התכנות. על שפות התכנות ידובר בהרחבה מאוחר יותר, כאן יצוין רק, שרוב שפות התכנות הנפוצות (PL/I, RPG, FORTRAN, COBOL) הן שפות עיליות (HIGH LEVEL LANGUAGES). כינוי זה מרמז, כי השפות הללו דומות לשפות, שבני האדם משתמשים בהן בחיי היומיום, השפה האנגלית, שפת המתמטיקה וכו'.

שפת תכנות, בה בולט הדמיון לשפה האנגלית ולמתמטיקה כאחת, הינה שפת PL/I. להלן דוגמה לתכנית קטנה ב-PL/I, המציגה דמיון זה.

```
TEMP : PROC OPTIONS (MAIN);
       DECLARE TOTAL, DAY;
       DECLARE PLACE CHAR (5),
               DATE CHAR (8),
               TEMPER FIX DEC (4,2);

READ : GET LIST (PLACE,DATE,TEMPER);
       TOTAL = TOTAL + TEMPER;
       DAY = DAY + 1;
       IF TEMPER = 9999 GOTO READ;
       AVG = TOTAL/DAY;
       PUT AVG;
       END TEMP;
```

ההוראות נכתבות על טופס קידוד, המתאים לשפה, בה נכתבת התכנית. בעבר, כל שורה של הטופס נוקבה בכרטיס נפרד, ולכן נגלה, כי מספר העמודות בכל שורה הוא 80. נתבונן בציור 5-9, בו רואים קטע תכנית בשפת COBOL, ראה גם ציור 7-3.

```
     PROCEDURE DIVISION.
     - - - - -
     - - - - -
 A1. READ RECORD1 AT END GOTO END1.
     MOVE CORRESPONDING IN-AREA TO OUT1.
     WRITE OUT1.
     GOTO A.
 END1.
```

ציור 5-9
- קטע תכנית ב-COBOL

קטע זה הינו למעשה חלק מתכנית קובול אשר נכתבה לפי התרשים שבציור 5-8.

שלב 4 - הזנת התכנית לזיכרון המחשב

לאחר שהשלים את הכתיבה, ידאג התכניתן לכך, שתכניתו תוזן לזיכרון המחשב. הוא יכול לעשות זאת באחת משתי דרכים:

1. התכניתן ימסור את התכנית להקלדה. קלדנית תקליד את התכנית על תקליטון, ודרך קורא התקליטונים תיקרא התכנית לתוך הזיכרון (שלב 2, ציור 5-5).

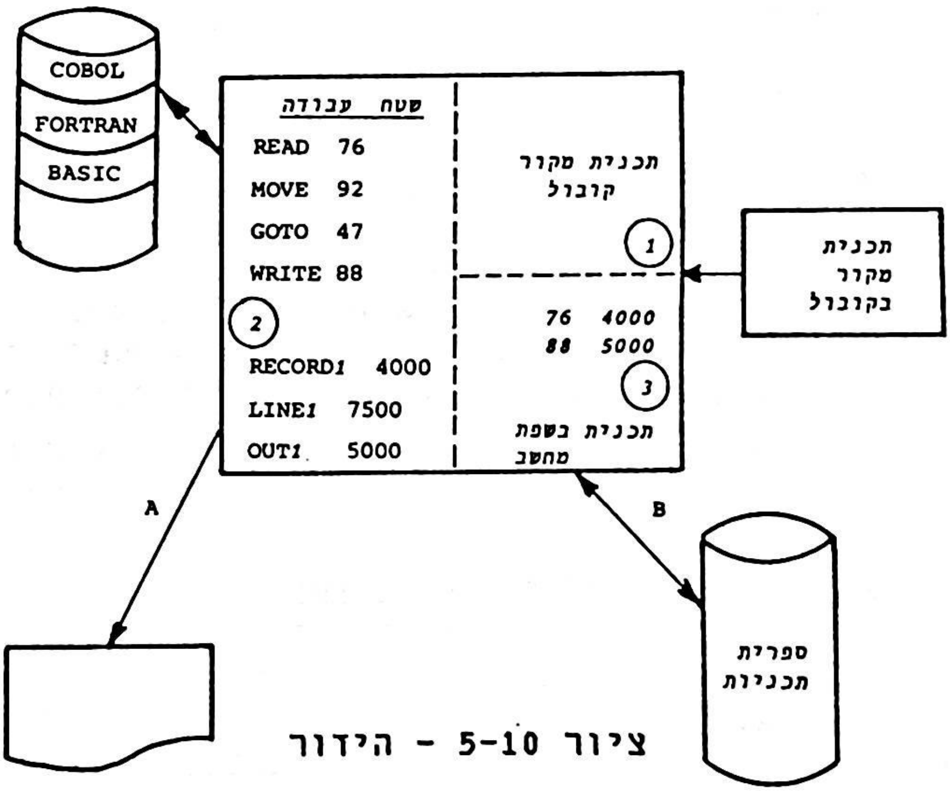
2. התכניתן יקיש את הוראות התכנית, באמצעות מסוף, וכך היא תוזן ישירות למחשב.

בשני המקרים יתוספו לפני התכנית ואחריה <u>הוראות פיקוח</u> (CONTROL STATEMENTS) מיוחדות, אשר מוסרות פרטים שונים על התכנית, שפת התכנות, קובצי עבודה וספריית התכניות.

שלב 5 - תרגום התכנית לשפת מחשב (הידור)

באחד מאמצעי האחסון החיצוניים (דיסק, או תקליטון של מחשב זעיר) נמצאים <u>מהדירים</u> (COMPILERS) לשפות התכנות השונות. אלו הן תכניות, המסופקות בד"כ על ידי יצרני המחשב, ותפקידן לתרגם את התכניות משפות התכנות העיליות לשפה, אשר מובנת למחשב.

שלב זה מורכב משני חלקים, אותם נתאר בעזרת ציור 5-10.



ציור 5-10 - הידור

1. <u>תרגום של תכנית המקור לשפת מכונה</u>

התכנית, שנכתבה בשפת תכנות נקראת <u>תכנית מקור</u> (SOURCE PROGRAM). בכל הוראה והוראה אפשר להבחין בפעלים, כגון קרא (READ), הדפס (WRITE) העבר (MOVE). כמו כן ניתן לזהות גם שמות (שהגדרתם ותיאורם מופיעים בקטע אחר של התכנית, שאינו מוצג כאן) כגון RECORD1, LINE1 וכו'. תכנית המקור מוזנת למחשב בצירוף עם הוראות פיקוח שונות. לפי הוראות הפיקוח נבחר המהדיר המתאים לתכנית המקור שהוזנה למחשב, והוא נטען בחלקו בזיכרון. בהמשך, נציג באופן כללי את פעולת המהדיר.

המהדיר, שהוא תכנית מחשב מורכבת מאד, מכיל טבלה, אשר מגדירה ערכים מספריים לפעלים של ההוראות, כמו למשל:

READ	76
MOVE	92
GOTO	47
WRITE	88

לאחר שתכנית המקור נקראת לזיכרון, מזהה המהדיר את השמות, ובונה מהם טבלה בשטח העבודה, המסומן בספרה 2 בתרשים 5-10. נניח שהטבלה נראית כך:

RECORD1	4000
LINE1	5000

עתה הכל מוכן למעשה לבניית התכנית בשפת המכונה. המהדיר מלקט את הערכים המספריים לפעלים של הוראות תכנית המקור, ואת הערכים המספריים של הכתובות, במקום השמות שבהוראות תכנית המקור.

במקום	אפשר לכתוב
READ RECORD1	76 4000
WRITE LINE1	88 5000

כך מקבלים תכנית, הכתובה כולה במספרים. למעשה המספרים אינם עשרוניים אלא בינריים, ובסופו של דבר התכנית תיראה בערך כך:

```
1011011001110010100
1000110011000011101
```

זוהי תכנית <u>בשפת מחשב או תכנית יעד</u> (OBJECT PROGRAM), אשר נבנית בקטע זיכרון, המסומן בספרה 3 בציור 5-10. בתוך כדי תרגום התכנית מזהה המהדיר שגיאות בתחביר. הכוונה לכתיב שגוי של הוראה (REED במקום READ) או לשימוש בלתי נכון בהוראה, כמו למשל חיסור אות ממספר.

2. הצגת הפלט של המהדיר

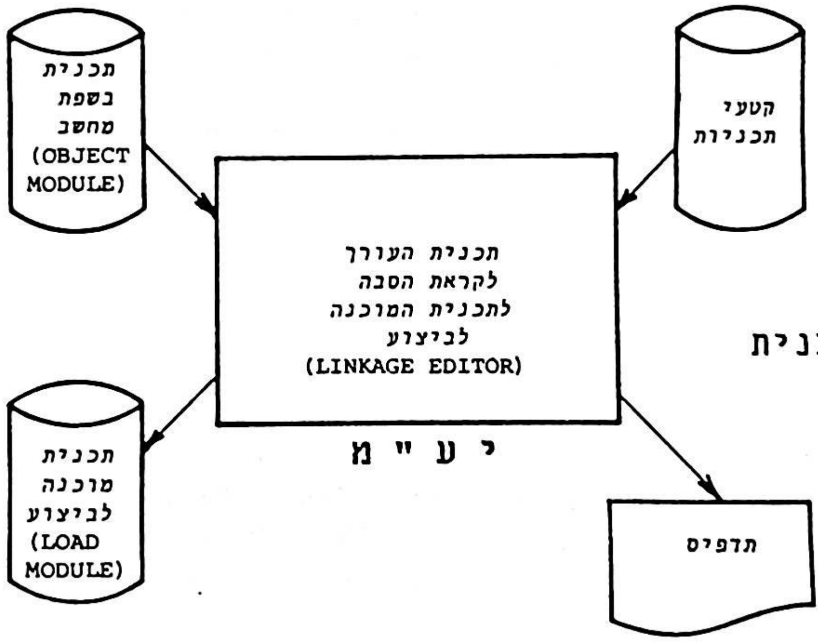
המהדיר, שהוא כאמור תכנית מחשב בפני עצמה, קולט את תכנית המקור, ובונה תכנית בשפת מחשב. הפלט של המהדיר הינו כפול. תכנית המקור מודפסת על המדפסת (ראה A ציור 5-10), כפי שהיא הוזנה לתוך הזיכרון. אם נמצאו שגיאות תחבירית, יציין המהדיר את השגיאות ואת מיקומן המדויק בתכנית. פלט זה יכול להיות מוצג גם על יחידת מצג.

מן הראוי לציין, <u>שהמהדיר אינו יכול לאתר ולגלות שגיאות לוגיות</u>, כגון: להוסיף מס הכנסה למשכורת הברוטו, במקום לנכותו ממנה. על איתור שגיאות לוגיות וניפוי כל סוגי השגיאות יוסבר בסעיף, המתאר את שלב 7.

אם תורגמה התכנית לשפת מחשב ללא שגיאות תחביר, אזי היא נכתבת גם על אמצעי אחסון זמני, בד"כ על דיסק או תקליטון. פלט זה יכול לשמש להרצה נסיונית של התכנית, המתוארת בשלב הבא.

אולם ייתכן שלקראת ההרצה יש לשלב בתכנית קטעי תכנות נוספים, שהוכנו קודם לכן. דוגמה לכך יכול לשמש קטע "דמי הבראה" של תכנית משכורת, אשר נכתב קודם לכן, ואשר מצורף לתכנית המשכורת הרגילה, נאמר פעמיים בשנה. בכדי לאפשר חיבור קטעים שונים של תכניות, חייבת התכנית בשפת מחשב לעבור שלב נוסף של <u>עריכה</u> (LINKAGE EDITING). בשלב זה קולטת תכנית מיוחדת, הנקראת LINKAGE EDITOR, את התכנית בשפת מחשב, ואם יש צורך, גם קטעי תכניות אחרים (MODULES). לאחר העיבוד מתקבלת כפלט תכנית מוכנה לביצוע, אשר נכנה אותה <u>תכנית נטענת</u> (LOAD MODULE). תהליך זה מוצג בציור 5-11.

מן הראוי לציין שתהליך זה הוא עקרוני. יתכן שבמחשבים שונים יהיו הבדלים בשלבי התהליך, או בשמות הפעולות והתכניות, להכנה של התכנית הנטענת.



ציור 5-11 - עריכת תכנית

י ע " מ

בשלב זה מתקבל גם פלט במדפסת. אם היו שגיאות בשלב זה, הן מצויינות בתדפיס.

<u>שלב 6 - הרצת ניסוי</u>

בשלב <u>הרצת הניסוי</u> (TEST RUN) עומדות בפני התכניתן שתי מטרות:

1. לגלות שגיאות לוגיות, שלא ניתנות לגילוי על ידי המהדיר (למשל לקבוע כמויות שליליות לפריטים במחסן).

2. לבדוק האם התכנית (שעד כה היתה רק סדרת הוראות כתובות בשפת תכנות) יכולה לבצע את המשימה לשמה היא נוצרה.

לשם מילוי משימות אלו מפעיל התכניתן את התכנית עם <u>נתוני בדיקה</u> (TEST DATA). נתונים אלו יכולים להיות חלק מרשומות הקלט; למשל, במפעל שבו 5000 עובדים מבצעים הרצת ניסוי של תכנית משכורת ובה 100-200 רשומות עובדים. ייתכן גם מקרה, בו ריצת הניסוי מתבצעת עם נתוני-דמה, שנוצרו אך ורק למען הריצות האלו.

בכל מקרה על התכניתן לדעת <u>מראש</u>, מהן התוצאות הצפויות, ולבדוק, שאמנם הן מתקבלות לאחר הפעלת התכנית. אם ניקח לדוגמה שגיאה לוגית, שהוזכרה קודם לכן - הוספת מס למשכורת, במקום ניכוי - הרי המשכורות תהיינה גבוהות הרבה יותר מן המצופה. זוהי שגיאה לוגית אופיינית, והמהדיר <u>אינו</u> מודיע עליה. התכניתן חייב למצוא בעצמו את מקום השגיאה בתכנית. איתור שגיאה לוגית יכול להמשך זמן רב במיוחד, כאשר התכנית ארוכה.

<u>שלב 7 - ניפוי שגיאות</u>

ניפוי שגיאות הינו אחד השלבים הפחות אהודים על ציבור התכניתנים. מאמצים רבים הושקעו בכתיבת התכנית, היתה הקפדה על מילוי כל הדרישות, והנה - קורה שהתכנית "אינה עובדת".

לחלק זה של העבודה מספר חלקים: בראש ובראשונה התכניתן מאתר את שני הסוגים של השגיאות (התחבירית והלוגית). לאחר מכן הוא מתקן בתכנית המקור את כל השגיאות, ז.א. כותב את ההוראות השגויות מחדש אך בצורה מתוקנת. לאחר מכן מוזנת התכנית למחשב, עוברת תרגום מחדש ומופעלת שוב עם נתוני בדיקה.

אם נתקבלו התוצאות המצופות, הכל שפיר, אך ייתכן שתתגלינה שגיאות נוספות. במקרה כזה יש לחזור שוב על שלבים 4-7 (אפשר לומר שזהו המקרה השכיח). תהליך זה של <u>ניפוי שגיאות</u> נקרא בשפת אנשי המחשב "DEBUGGING" או בתרגום חופשי "הוצאת כל המקקים מן התכנית".

שלב 8 - ביצוע התכנית

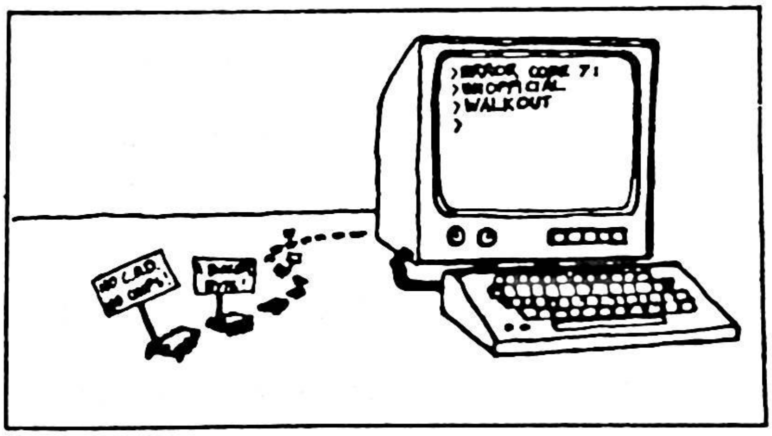
שלב זה, הינו האחרון בין שלבי התכנות, ומהווה, לכאורה, את הסיום של תהליך התכנות כולו. התכנית נקייה משגיאות, והיא יכולה לבצע את המשימה לשמה היא נוצרה. תכנית משכורת תקלוט את קובץ כוח האדם ותפיק תלושי שכר, תכנית לניהול מלאי תקלוט את קובץ הפריטים ותפיק דו"ח יתרות מלאי ודו"ח הזמנות על פי רמות מלאי מינימום.

אולם יש לציין, שתהליך עיבוד נתונים לפתרון בעיה מסוימת מורכב בדרך כלל ממספר מהלכים או תכניות. לאחר בדיקת כל התכניות בנפרד יש להריץ רצף שלם של תכניות, החל מהתכנית הראשונה, המקבלת את הנתונים, ועד לתכנית האחרונה, אשר מפיקה את הפלט הסופי.

בתהליך עיבוד קובץ יתרות עו"ש בבנק יכולים להיות המהלכים הבאים: תכנית קלט של תנועות (משיכות והפקדות), תכניות מיון, תכנית עדכון של קובץ יתרות ותכנית להדפסת דפי חשבון. הפלט של המהלך הראשון יהיה הקלט של המהלך השני הבא אחריו, וכך הלאה עד להפקת הדו"ח הסופי.

עוד נקודה חשובה הראויה לציון: התכניתן כתב תכנית משכורת, ניפה את כל השגיאות, והתכנית הדפיסה תלושי משכורת. חישוב המשכורת הבאה יחל בעוד שלושה שבועות. האם תשאר התכנית בזיכרון המחשב? ודאי שלא.

כל תכנית תקינה, שנבדקה ונמצאה תקינה, <u>מקוטלגת</u> (נכתבת) בספריית התכניות, שנמצאת על הדיסק (ראה ציור 5-10). קטלוג התכניות אינו בתחום אחריותו של התכניתן ולכן ידובר על כך בפרק 8, כשנדון במערכות ההפעלה.



הרדעה: ה"ג'וקים" נוטשים

פרק 5 - שאלות

1. הגדר את המושגים הבאים:

   א. חמרה

   ב. תכנה

2. ביחידת מחשב גדולה חמישה סוגים שונים של תפקידים. האם ניתן לוותר, לדעתך, על אחד מהתפקידים? נמק

3. מהם חמשת השלבים העיקריים בניתוח מערכות?

4. מה קודם למה, תרשים זרימה לתרשים תכנות, או להיפך? נמק.

5. אלגוריתם מהו?

6. מצא את השגיאות באלגוריתם הבא:

   א. לך למכולת.

   ב. קח כסף.

   ג. קנה ככר לחם, ליטר חלב ועשר ביצים.

   ד. קח איתך סל.

   ה. הבא גם עיתון.

   ו. חזור הביתה עד השעה אחת.

7. נניח, שקרוביך מגיעים ארצה לביקור משפחתי, זו הפעם הראשונה. הם מבינים עברית, אך אינם מכירים את כללי השימוש בטלפון ציבורי בארץ. הכן תרשים, בו תנחה את קרוביך כיצד לבצע שיחה באמצעות טלפון ציבורי.

8. הסבר בקצרה מה זה הידור?

9. מדוע חשוב כל כך להריץ תכנית עם נתוני בדיקה (TEST DATA)?

10. בשלבים שונים של עבודת התכניתן מתקבלים דו"חות מודפסים. ציין ותאר אותם.



ציור 5-12 - מחשב עסקי ליד חדר מחשב מרכזי

פרק 6 – טכניקות תכנות

כתיבה של תכנית מחשב הינה פעולה קשה ומסובכת, וצופנת אפשרויות רבות לשגיאות. ככל שהמחשב משוכלל יותר, וככל שדרישות הביצוע מורכבות יותר, כך נעשית כתיבת התכנית ממושכת ויקרה יותר. בגלל סיבה זו פותחו טכניקות תכנות שונות, כגון תרשימים וטבלאות החלטה. השימוש בטכניקות הללו יכול לפשט תהליכים מורכבים ומסובכים, ולאפשר את התכנון בצורה טובה יותר. הבה נמחיש זאת במעשה שהיה.

בקיץ שעבר התארח בביתי ג'יסי עם משפחתו. ג'יסי הוא תושב ארה"ב, אך כל קיץ מבלה את חופשתו בארץ עם אשתו ושני ילדיו. יום לאחר שהגיעו לישראל, כתבו הילדים מכתב לחבריהם בארה"ב. ג'יסי קרא לבנו ואמר: "דני, בקצה הרחוב ישנו משרד דואר, גש לשם, ושלח את המכתב". דן בן השמונה שמח, שמטילים עליו משימה כה חשובה, לקח את המכתב, ופתח בריצה אל הדואר. כעבור חמש דקות חזר: "אבא, הם רוצים כסף ישראלי" - אמר. מסתבר, שדני הקטן לקח איתו כסף אמריקאי. עכשיו נטל שטר של 100 שקלים, אך הפעם שכח את המכתב בבית, ושוב חזר בלי לשלוח את המכתב. כשהלך בפעם השלישית - היה כבר משרד הדואר סגור. דני המסכן פרץ בבכי מר.

כדי לנחם אותו, עשינו שני דברים: נתנו לו חפיסת שוקולד, ואח"כ ישבנו איתו, וכתבנו על דף נייר, מה צריך לעשות:

1. לקחת את המכתב,

2. לקחת שטר של 100 שקלים,

3. ללכת אל סניף הדואר,

4. אם סניף הדואר פתוח, לקנות בול,

5. להדביק את הבול על המטעפה,

6. לשלשל את המכתב בתיבה,

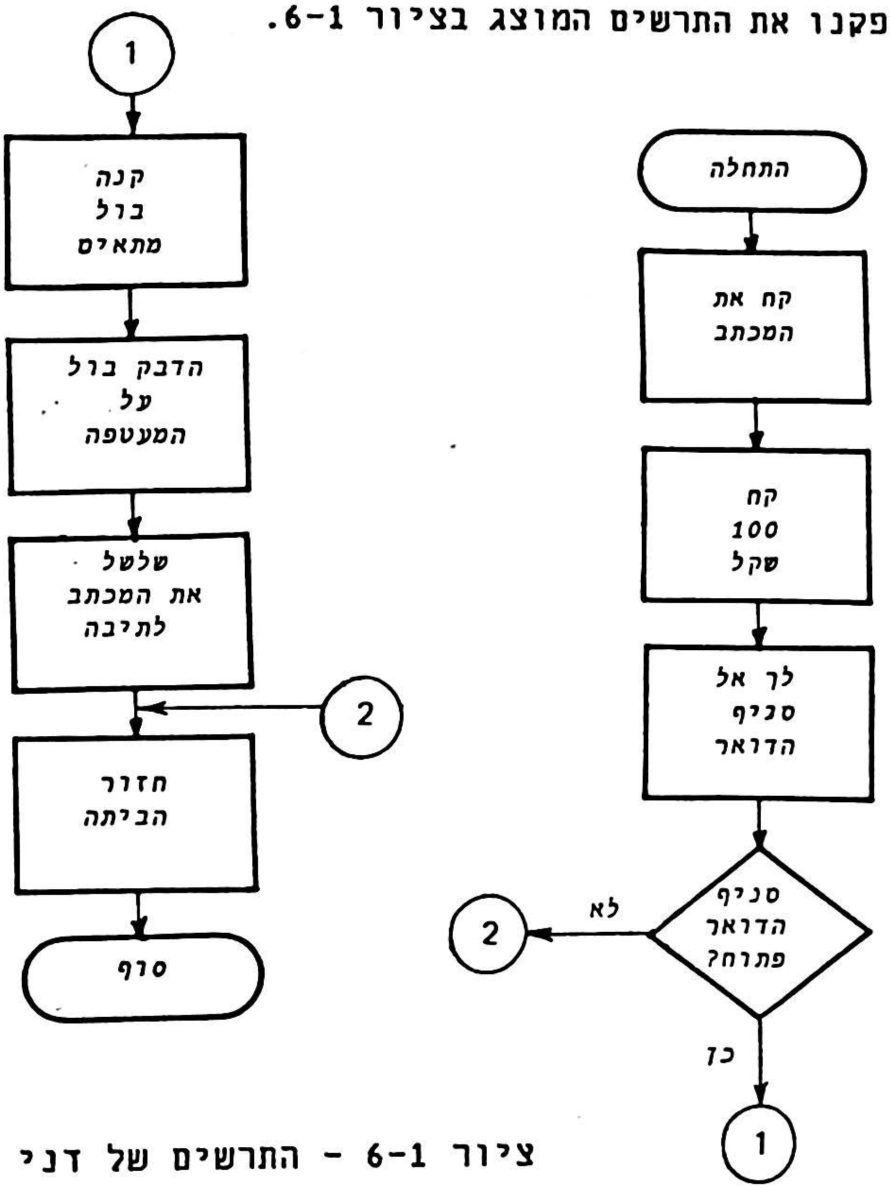
7. לחזור הביתה.

לפתע שאל דני: "ומה לעשות, אם סניף הדואר יהיה סגור?" אז ראינו, שכדאי לעשות תכנון שלם יותר, ואולי להשתמש בתרשים גרפי, כדי לראות אם לא שכחנו פעולה כלשהי.

לפני שהתחלנו לשרטט את התרשים, קבענו לעצמנו, שכל פעולה, שעל

דני יהיה לבצע תירשם במלבן. את ברירת המצב של סניף הדואר, אם פתוח או לא, נרשום בתוך מעוין. בנוסף לכך נסמן בצורה בולטת, היכן מתחיל השרטוט, ובאיזו נקודה הוא מסתיים.

לבסוף הפקנו את התרשים המוצג בציור 6-1.



ציור 6-1 - התרשים של דני

עד כאן המעשה. עתה נראה מהם הכללים לשימוש בתרשים מלבני.

6.1 תרשים מלבני

<u>תרשים מלבני</u> (BLOCK DIAGRAM) שימושי מאוד בתחומים רבים, בהם נדרש לפתור בעיה כלשהי, שמאפייניה הם:

- הפתרון תלוי בגורמים רבים,
- ישנן דרכי פתרון שונות,
- יש מצבי ברירה, ויש להחליט באיזו דרך לבחור.

לתרשים מלבני יש מספר כללי כתיבה:

1. לכל תרשים יש התחלה ויש סוף, המסומנים בדרך-כלל בתוך תבנית אליפסה.

2. כל פעולה נכתבת במלבן.

3. במלבן אחד נוהגים לכתוב פעולה אחת בלבד.

4. שאלות, משפטי תנאי או מצבי ברירה רושמים במעוין.

5. הצורות ההנדסיות השונות קשורות ביניהן בחצים.

6. ההתקדמות בתרשים היא מן ההתחלה אל הסוף לפי החצים (מלמעלה למטה ומימין לשמאל, או משמאל לימין).

7. מכל מלבן יוצא חץ אחד בלבד.

8. מכל מעוין יוצאים שני חצים, חץ אחד מורה על אפשרות, בה התשובה חיובית לשאלה המסומנת בתוך המעוין, והחץ השני מצביע על התשובה השלילית.

9. ניתן לחזור על חלקים של התרשים מספר פעמים בלתי מוגבל.

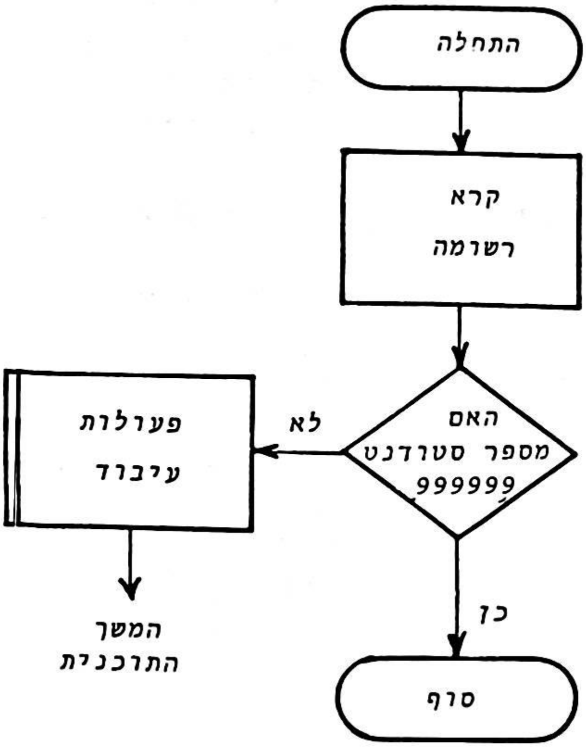
10. מותר לעבור ממעוין אחד למעוין אחר.

<u>תרשימים לביצוע פעולות עיבוד פשוטות</u>

בדוגמאות הבאות נניח תהליכי חישוב פשוטים אשר מהוים קטע ממערך תכנית מורכב יותר. הבעיות שתוצגנה מיועדות להדגים את הבנייה הלוגית של תהליך העיבוד.

בפעולות קריאה של רשומות הנתונו, שהתכנית הסטנדרטית של המחשב, המפקחת על ביצוע פעולות הקריאה, מזהה את הרשומה האחרונה (במבנה סטנדרטי), המציינת את סוף הקובץ, ואשר אינה מכילה נתונים כלשהם. לפיכך ביצענו סיעוף <u>מתוך</u> הוראת הקריאה.

כאשר הרשומה האחרונה בקובץ אינה במבנה סטדנרטי של המערכת, יש לבדוק אותה במפורש. נניח, למשל, שבקובץ סטודנטים, מכילה הרשומה האחרונה את המספר 999999 בשדה מספר סטודנט. התרשים מוצג בציור 2-6.

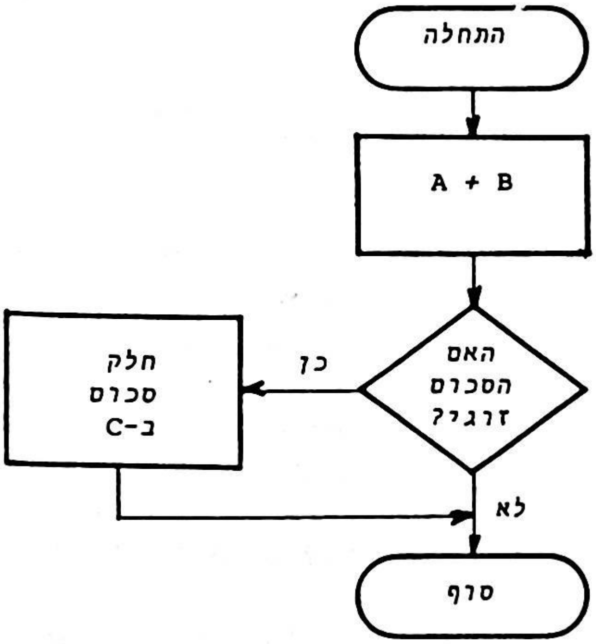


ציור 6-2

- תרשים לבדיקת רשומה אחרונה

<u>בעיה אי - חיבור שני מספרים והתניה</u>

יש לחבר שני מספרים A ו-B. אם הסכום אינו זוגי, סיים את העבודה; אם הסכום הוא זוגי, חלק את הסכום ב-C וסיים. ראה ציור 6-3.



ציור 6-3

- תרשים לבעיה אי

התרשים הנכון הוא תרשים, המטפל בכל תנאי הבעיה. כאשר ניסוח התנאים והפעולות מדויק ובהיר, ומספר הצעדים הנדרשים הוא הקטן ביותר, אפשר גם להגדירו כתרשים טוב.

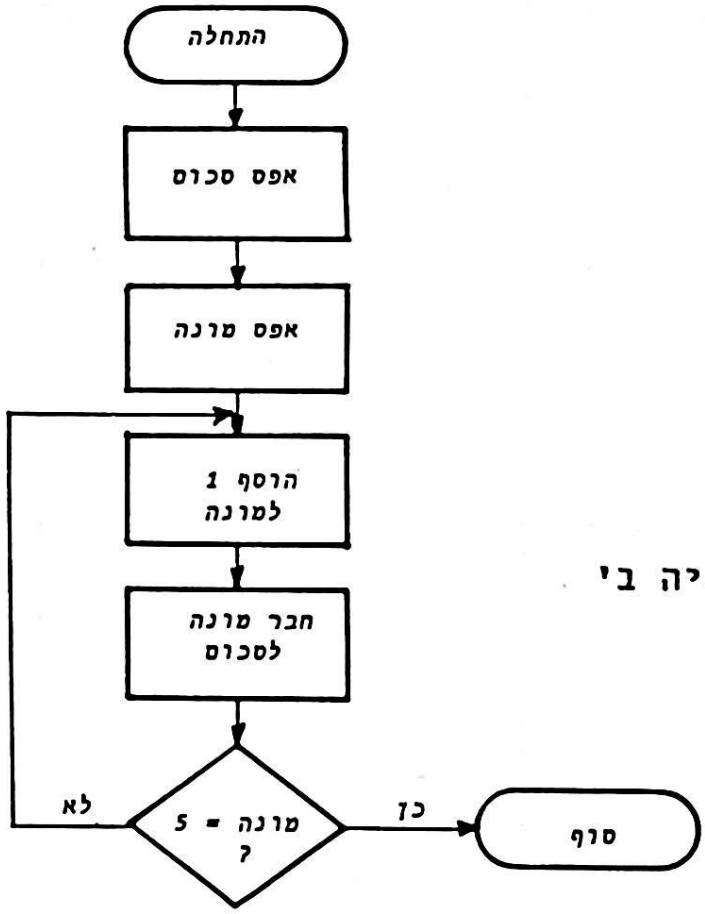
בעיה ב׳ - סיכום של סדרת מספרים בלולאה

1. תיאור הבעיה

יש לסכם את כל המספרים מ-1 עד 5. פעולה זו דורשת 5 פעולות חיבור של הערכים 1 עד 5 בהתאמה. כדי לשלוט על מספר הפעולות הנדרשות יש לבקר זאת בעזרת מונה, אשר יורה על סיום, כאשר ערכו יגיע ל-5.

כאשר מסכמים למונה, יש לאפס אותו לפני תחילת הפעולה. פעולה של איפוס מונה נהוגה במכונות סיכום פשוטות, וכך יש לנהוג גם במחשב.

2. התרשים המלבני



ציור 4-6 - תרשים לבעיה ב׳

המונחים "מונה" ו-"סכום" מציינים את הפעולה, הנדרשת על ידנו. למעשה בשני המקרים מבוצעת פעולת סיכום למונה, או בשפת המחשב - <u>פעולת חיבור למשתנה</u>.

3. בדיקת התרשים

כדי לבדוק, אם התרשים עונה על תנאי הבעיה, יש לבדוק אותו בצורה מסודרת בעזרת <u>נתוני ניסוי</u>.

בבעיה שלפנינו אין נתוני קלט ועל כן נתחיל מיד בתהליך העיבוד המוצג בתרשים. לצורך הבדיקה נכין טבלה, שתכיל בצד אחד את הפעולה המבוצעת ובצידה השני את מצב המונה והסכום.

ה פ ע ו ל ה	מ ו נ ה	ס כ ו ם
1. אפס סכום		0
2. אפס מונה	0	
3. הוסף 1 למונה	1	
4. חבר מונה לסכום		1
5. האם מונה = 5 ? - לא		
6. הוסף 1 למונה	2 (=1+1)	
7. חבר מונה לסכום		3 (=1+2)
8. האם מונה = 5 ? - לא		
9. הוסף 1 למונה	3 (=2+1)	
10. חבר מונה לסכום		6 (=3+3)
וכו'		
- חבר 1 למונה	5 (=4+1)	
- חבר מונה לסכום		15 (=10+5)
- האם מונה = 5 ? - כן		==
==		
סוף, התוצאה הנכונה 15		
==		

<u>בעיה ג' - הכנה של דו"ח מודפס</u>

בדוגמה הבאה נתייחס שוב אל בעית מיכון הספרייה, אשר הועלתה בפרק 5. לפנינו קטע מתיק התכנון, המטפל בהכנת דו"ח מודפס על כל הספרים, שהוזמנו והגיעו אל ספרייה עירונית כלשהי.

ג. תיאור הבעיה

א. **קלט:** אל הספרייה של העירייה מגיעים כל הספרים החדשים, כאשר לכל אחד מהם מצורף כרטיס מנוקב. בספרייה שלושה מדורים: ספרות יפה, ספרי ילדים, ספרי מדע.

תיאור כרטיס/רשומת הקלט ניתן להלן:

תאריך	מחיר	שם המו"ל	שנת הוצאה	שם המחבר	שם הספר	קוד הזמנה

ב. **העיבוד:** יש לקרוא את הכרטיסים, להדפיס דו"ח שבועי לפי מבנה, הנתון להלן. יש להעלות את רשומת הקלט על סרט מגנטי לצורך העיבוד החודשי (חיוב הספרייה, זיכוי בהנהח"ש וכדי).

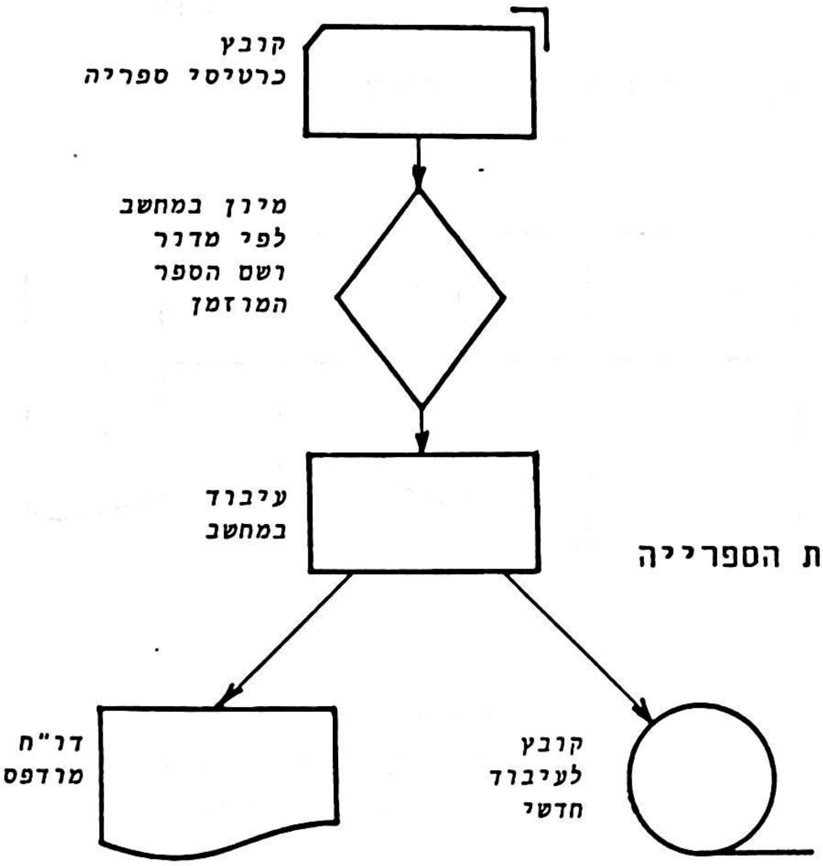
בתהליך העיבוד המוצג כאן אין מיון קודם של הקלט, לפיכך יודפס הפלט לפי סדר רשומות הקלט. הסיכומים יינתנו בסוף הדו"ח במרוכז.

ג. **פלט:** מבנה הדו"ח המודפס ניתן להלן:

```
------------------------------------------------------------------
|                                                                |
|               ספריות עירוניות - דו"ח שבועי                     |
|                                                                |
|----------------------------------------------------------------|
|                                                                |
| מדור בספרייה      שם המחבר        שם הספר       תאריך הזמנה   |
|                                                                |
|----------------------------------------------------------------|
|                                                                |
|   --------        ------         -------        -------       |
|   --------        ------         -------        -------       |
|   --------        ------         -------        -------       |
|                                                                |
------------------------------------------------------------------
```

בסוף הדו"ח יש להדפים את סה"כ הספרים, שנתקבלו בעבור כל מדור בספרייה. אם חסר קוד מדור בכרטיס, יש להדפיס כוכביות בשדה המדור ולהוסיף ציון שגיאה בסוף הדו"ח, למשל "שגוי - חסר שם מדור".

2. <u>תרשים הזרימה</u>



ציור 5-6
- תרשים זרימה לבעית הספרייה

3. <u>התרשים המלבני</u> יציג את שלבי העיבוד במחשב, הפקת הדו"ח והסרט לעיבוד החודשי. יש להדגיש את הנקודות הבאות:

א. <u>מונים:</u> בתחילת העיבוד אפשר להגדיר שדות, שימלאו תפקיד של מונה. לפי נתוני הבעיה אפשר גם לדאוג לערך התחלתי, שמונים אלה יכילו. בדוגמה שלפנינו המונים יאופסו בתחילת העיבוד. יש לתת לכל מונה שם מיוחד, שונה מכל שם אחר בתכנית.

ב. <u>עריכה:</u> כאשר הפלט הוא דו"ח מודפס, יש לערוך את השדות השונים, כדי שהשורה תהיה נוחה לקריאה, ותכיל בדיוק את הנתונים שרוצים להדפיס (לא פחות, אך גם לא יותר). העברת שדות שונים לשדות הפלט וסידורם עם מרווחים מתאימים - נקראת עריכה.

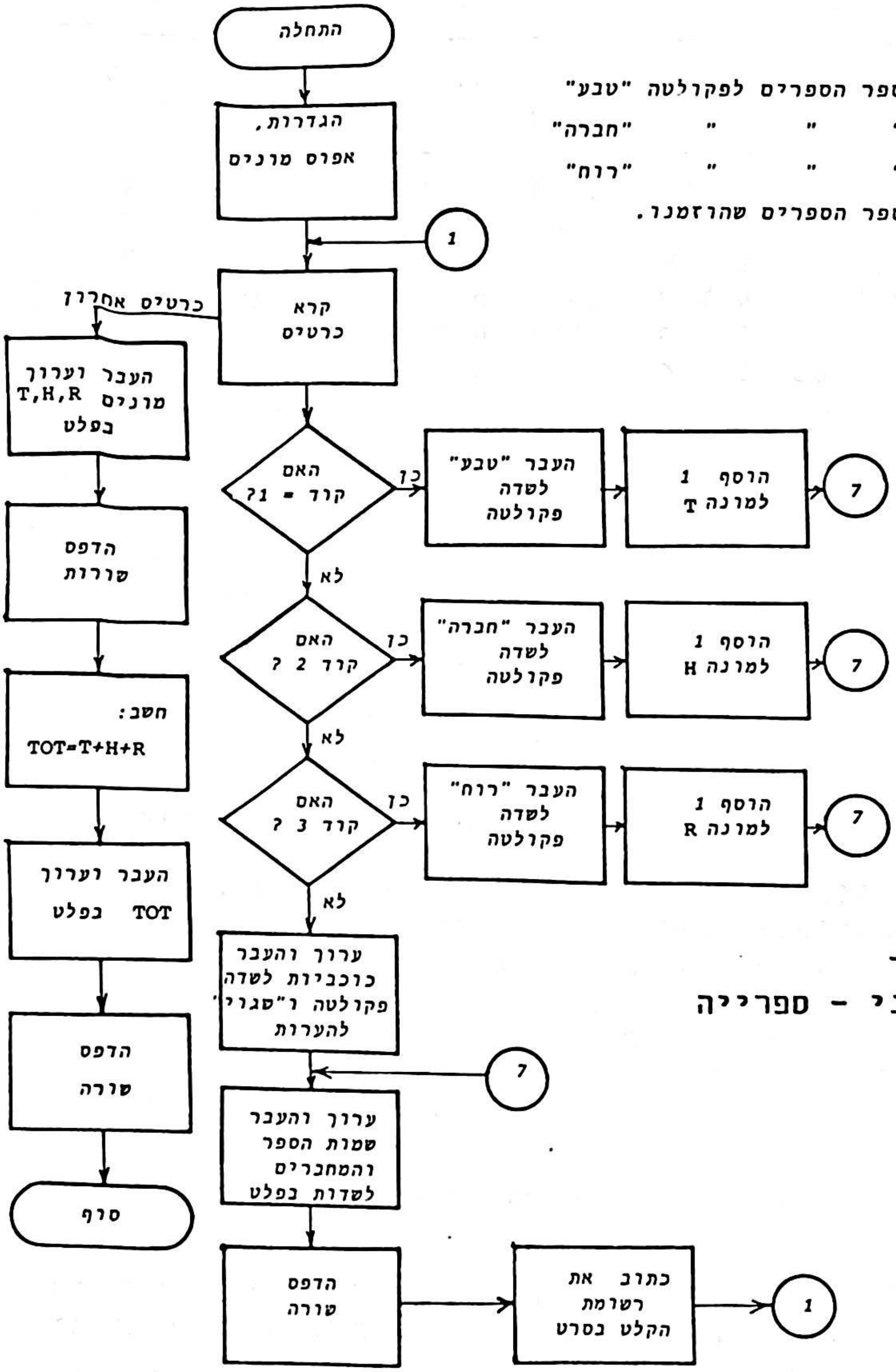
ג. בפתרון שלפנינו התעלמנו מבעית הכותרת ומספור הדפים.

מקרא:

M - מונה מספר ספרים למדור "מדע"
S - מונה מספר ספרים למדור "ספרות יפה"
Y - מונה מספר ספרים למדור "ספרי ילדים"
TOT - סה"כ הספרים שהוזמנו.

<u>מקרא:</u>

T - מונה מספר הספרים לפקולטה "טבע"
H - " " " "חברה"
R - " " " "רוח"
TOT - סה"כ מספר הספרים שהוזמנו.



ציור 6-6 -
תרשים מלבני - ספרייה

4. <u>בדיקת התרשים המלבני</u>

כדי לבדוק אם התרשים, או הפתרון המוצג ע"י התרשים, עונה על תנאי הבעיה, משתמשים בתחילה בנתוני ניסוי.

בבעיה המוצגת ישנן פעולות עיבוד, המבוצעות עם נתוני קלט. נתוני הפלט כוללים דו"ח מודפס, שגם אותו יש לבדוק לפרטיו.

דוגמה לנתוני קלט לבדיקה:

קוד מדור	שם הספר	המחבר	שנת הוצאה	שם המו"ל	תאריך הזמנה	מחיר
1	אב 1	מחבר 1	1981	מול 1	1.8.83	2615.-
3	אב 2	מחבר 2	1983	מול 2	5.8.83	1530.-
2	אב 3	מחבר 2	1980	מול 2	6.7.83	1812.-
1	אב 4	מחבר 5	1980	מול 4	8.8.83	3014.-
4	אב 5	מחבר 6	1983	מול 1	9.8.83	915.-
3	אב 6	מחבר 3	1983	מול 3	9.8.83	710.-

<u>דוגמה לדו"ח הצפוי בתהליך העיבוד:</u>

המדור בספרייה	שם המחבר	שם הספר	מחיר
מדע	מחבר 1	אב 1	2615.-
ילדים	מחבר 2	אב 2	1530.-
ספרות יפה	מחבר 2	אב 3	1812.-
מדע	מחבר 5	אב 4	3014.-
******	מחבר 6	אב 5	915.-
ילדים	מחבר 3	אב 6	710.-

חיוב למדור ספרי מדע	סה"כ	5629.-	שקלים
חיוב למדור ספרות יפה	סה"כ	1812.-	שקלים
חיוב למדור ספרי ילדים	סה"כ	2240.-	שקלים
	סה"כ	9681.-	שקלים

1 שגיאה בקוד מדור.

<u>6.2 תכנות מבני</u>

כפי שאפשר היה להיווכח בסעיף הקודם, לבעיה מסוימת אחת ניתן היה למצוא כמה וכמה דרכי פתרון אפשריות בעזרת הצגה גרפית של הפתרון. להצגה גרפית כזאת ישנם מספר יתרונות:

1. ניתן להציג בצורה נוחה וברורה את הקטעים העיקריים של התכנית ואת הקשרים ביניהם.

2. ניתן לראות את הרצף של הפעולות העיקריות בכל קטע של

התכנית בנפרד, ואת רצף הקטעים העיקריים המרכיב· התכנית כולה.

3. הצגה גרפית כזו מאפשרת לאנשים נוספים קריאה והבנה של התכנית, בלי להכנס לכל פרטיה.

אולם "אליה וקוץ בה". במשך עשרות שנים נהגו תכניתנים לפתח כללי עבודה משלהם ביצירת פתרונות לוגיים בתכניותיהם, אך ככל שהיתה התכנית גדולה - כך הלך והסתבך המבנה בשל הסתעפויות רבות, תוספות שונות, תיקונים (כתוצאה מהרצת התכנית) ועדכונים שונים. כל אלה גרמו לכך, שלעתים לא התמצא התכניתן עצמו בתרשים המלבני, שיצר אך לפני זמן לא רב.

אחזקת התכניות הללו ועדכונן מידי פעם נתקלו בקשיים. כאשר עזב התכניתן את מקום עבודתו, ומישהו אחר היה צריך לטפל בתכנית שלו, היה צורך "לפענח" את התכנית שכתב, פעולה קשה במיוחד אם התכנית גדולה ומורכבת. מכאן נבע הצורך להכניס סדר ושיטה בתחום זה של עיבוד נתונים. אנשי מדע מתחום המחשבים והמתמטיקה פיתחו שיטות, אשר אפשרו ליצור תכניות בעזרת כלים וסטנדרטים קבועים, כדי שהטיפול בתכנית יהיה קל יותר, ואחזקתה לא תיתקל בקשיים.

אחת מן השיטות שפותחו נוגעת לתחום התרשימים ולכתיבת התכניות, והיא נקראת <u>תכנות מבני</u> (STRUCTURED PROGRAMMING). השיטה מתבססת על תיאורמה מתמטית, הקובעת, שניתן לכתוב כל תכנית בעזרת שלושה מבנים בסיסיים מוגדרים מראש.

בשיטת התרשימים בתכנות מבני קיימים מספר כללים:

1. לכל התרשימים מבנה קבוע, שאין לסטות ממנו.

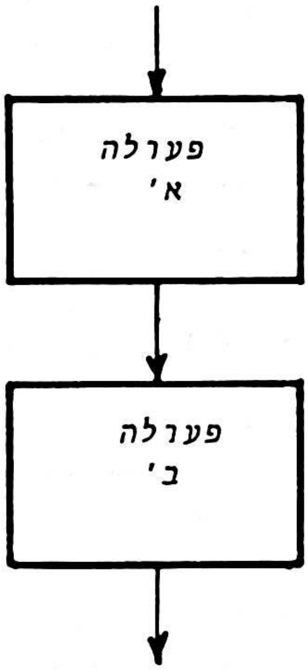
2. לכל מבנה נקודת כניסה אחת ונקודת יציאה אחת.

3. חלקי התכנית ניתנים לחלוקה מפורטת יותר, כפוף לכללים 1 ו-2.

נסקור עתה את שלושת המבנים הבסיסיים ואת המבנה הנוסף:

1. <u>מבנה סדרתי (SEQUENCE)</u>

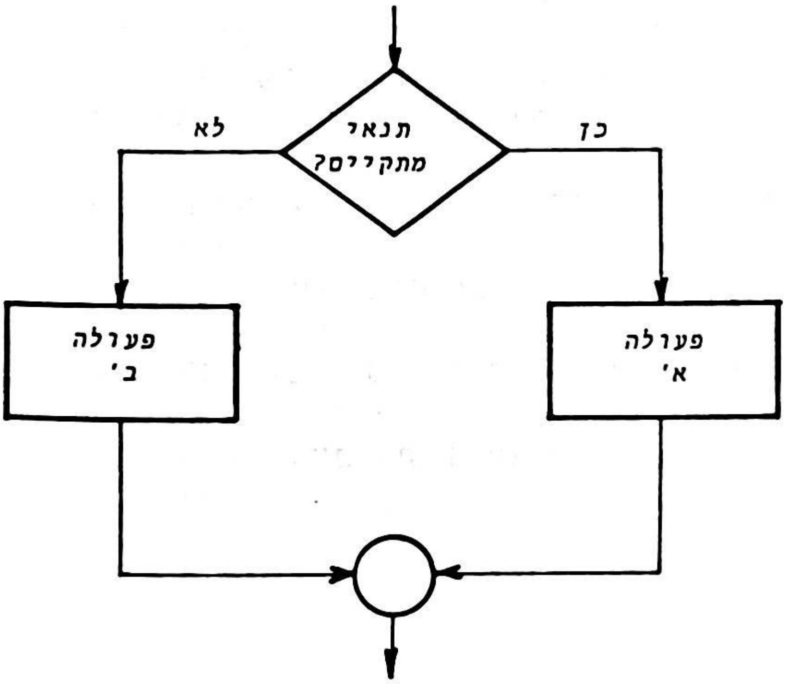
במבנה זה כל פעולה נרשמת במלבן, וסדרת הפעולות מתוארת ע"י שורת מלבנים המחוברים ביניהם בחצים.



ציור 6-7 - מבנה סדרתי

2. <u>מבנה תנאי "אם-אז" (IF-THEN-ELSE)</u>

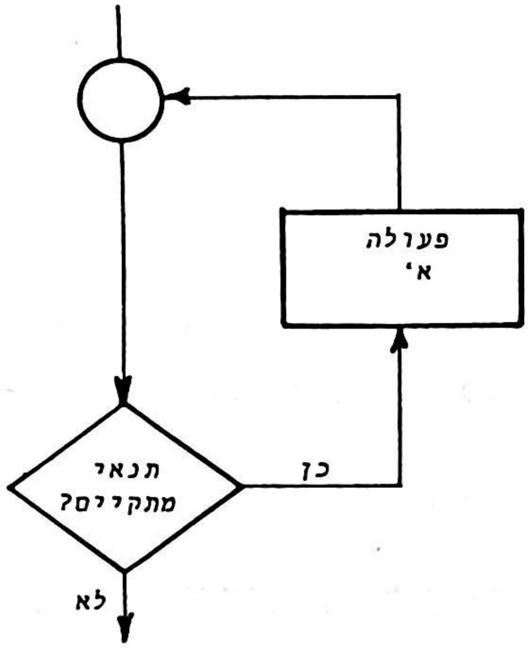
מבנה זה מאפשר הסתעפות מותנית לאחת משתי פעולות וחזרה למסלול הראשי, כפי שמוצג בציור 6-8.



ציור 6-8 - מבנה תנאי

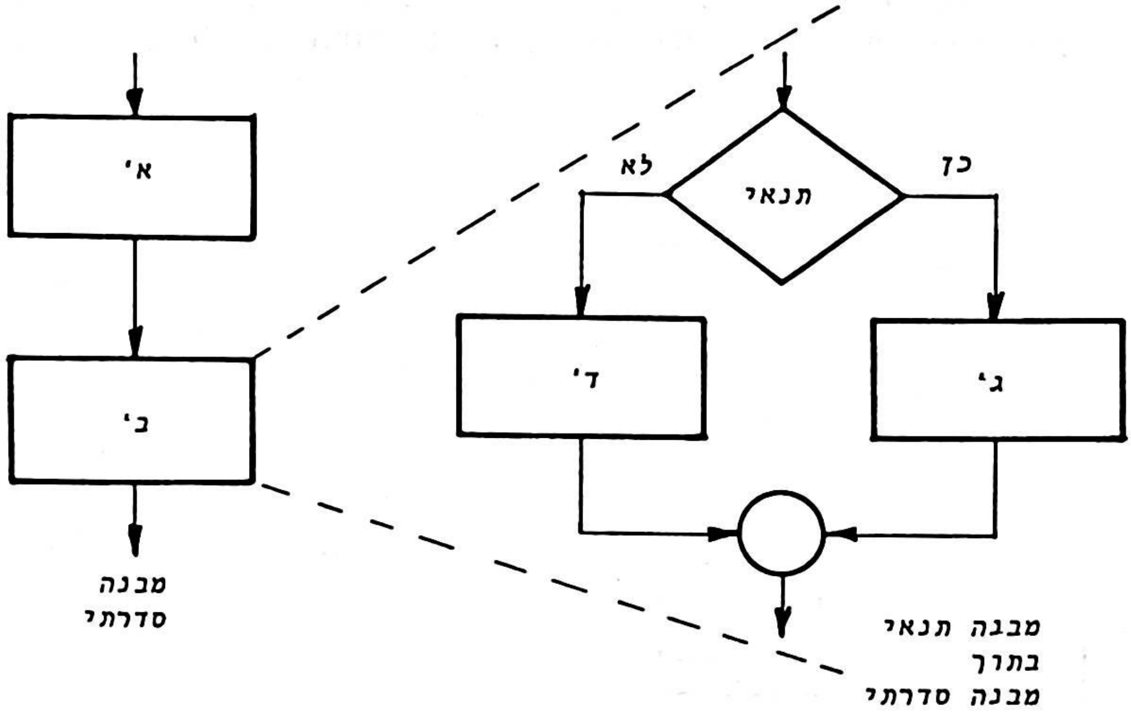
3. <u>לולאה "עשה כל עוד" (DOWHILE)</u>

פעולה A מתבצעת כל עוד התנאי, הנבדק במעוין, מתקיים.



ציור 6-9 - מבנה לולאה

פעולה ב בתרשים במבנה סדרתי כלשהו ניתנת לפירוק כפי שמוצג בציור 6-10.



ציור 6-10 - פירוק למבנים נוספים

4. <u>לולאה "עשה עד ש..." (DOUNTIL)</u>

ניתן ליצור מבנה נוסף לשלושת המבנים הבסיסיים, אשר כפוף לכללים הקודמים.
פעולה א תתבצע בצורה מחזורית, עד שתנאי מסוים יתקיים, למשל מונה יגיע ל-100.

צירופי תנאים

		רשימת תנאים	1	2	3	4	5	6
אם ... (תנאי)	1							
	2							
	3							
	4							
		רשימת פעולות						
אז ... (פעולה)	1							
	2							
	3							
	4							
	5							

ציור 13-6 - מבנה של טבלת החלטה

טבלת החלטה מורכבת מארבעה חלקים:

בחלקה העליון של הטבלה נרשמים <u>התנאים</u>, בהם תלוי ביצוע או אי-ביצוע של פעילות מסוימת, ובחלק התחתון תירשמנה <u>הפעולות</u>, שיש לבצע. בחלק השמאלי העליון נרשמים התנאים, כל תנאי בשורה נפרדת. בחלק הימני העליון נרשמים <u>צירופי תנאים</u>, אשר מוצגים על ידי מספרים. ניתן לראות בחלק זה תשובות, אשר ניתנות לשאלות, שנשאלו בחלק השמאלי העליון, כמו למשל: "האם זהו סוף קובץ? אם כן, הדפס שורת סיכום".
משבצות ריקות מציינות תנאי או פעולה שאינם רלבנטיים.

כדי להמחיש את המבנה ואת השימוש בטבלת החלטה מוצגת דוגמה בציור 14-6, שבה מברר תלמיד בית-ספר לעצמו, מה ללבוש לפני צאתו לבית-הספר, בהתאם לתנאי מזג האויר.

			1	2	3	4
אם ... (תנאי)	1	יורד גשם, אך לא קר.	כן	לא	לא	לא
	2	קר, אך בהיר		כן	לא	לא
	3	קר, מעונן			כן	לא
אז ... (פעולה)	1	ללבוש בגדים חמים		X	X	
	2	ללבוש בגדים רגילים				X
	3	ללבוש מעיל גשם	X			
	4	ללבוש רק סודר	X			
	5	לקחת מטרייה	X		X	

ציור 14-6 - טבלת החלטה לבעית "מה ללבוש"

את טור מס' 1 ניתן לקרוא כך:
<u>אם</u> יורד גשם, אך לא קר, <u>אז</u> לוקחים סודר, מעיל גשם ומטרייה.

את טור מס' 2 ניתן לקרוא כך:
<u>אם</u> לא יורד גשם, אך קר, <u>אז</u> יש ללבוש בגדים חמים בלבד.

ננסח עתה את הכללים למילוי טבלאות החלטה:

1. <u>רשימת התנאים</u> - התנאים ייכתבו בחלק השמאלי העליון של הטבלה. רצוי שכל שורה תכיל רק תנאי אחד כמו למשל "משכורת קטנה מ-280,000 שקל" או "מספר ילדים גדול מ-0".

2. <u>רשימת פעולות</u> - נכתבת בחלק השמאלי התחתון של הטבלה. רצוי למספר את הפעולות ולסדר אותן לפי סדר הגיוני, אך אין זה הכרחי. לעתים נדרשים לחזור על כתיבת פעולה מסוימת פעמים נוספות, כדי להבטיח בהירות ושלמות בקבוצת פעולות מסוימת.

3. <u>צירופי התנאים</u> - נרשמים בחלק הימני העליון של הטבלה, בעמודות ממוספרות. נוהגים לכתוב את המלים "כן", "לא" בשורות המתאימות לכל צירוף, אך יש ורושמים "1" כדי לסמן קיום תנאי מסוים ו-"0" לאי-קיום התנאי. בצירופים, שבהם לתנאי כלשהו אין משמעות, משאירים מרווח (BLANK), או רושמים את סימן המינוס.

4. <u>צירופי פעולות</u> - נכתבים או מסומנים בחלק הימני התחתון של הטבלה. נוהגים לסמן בסימן X את הפעולות, שיש לבצע, בהתאם לצירופי התנאים שמעליהם בטבלה. הצירוף הינו אוסף של פעולות אחדות, שיש לבצע, כאשר מתקיים צירוף של תנאים כלשהם.

לשם סיכום, נציג עתה לדוגמה בעייה בניהול מחסן פריטים ופתרונה בשלש דרכים, בשיטת תרשים מלבני, שיטת תרשים מבני ובשיטת טבלת החלטה.

<u>בעיה בניהול מלאי</u>

ניקח לדוגמה קטע מתכנית גדולה יותר, אשר מטפלת בניהול מלאי במחסן. קטע זה חייב לבדוק, אם ברשומת הקלט מופיע תוכן תקף בשדה "כמות במחסן" ובשדה "כמות מינימום". אם תנאי זה מתקיים, יש לבדוק, אם הכמות במחסן נמוכה מן המינימום הדרוש, ואז יש להדפיס הזמנה. במקרה שהכמות במחסן גדולה מן המינימום הדרוש, עוברים לרשומה הבאה. במקרה שבאחד השדות לפחות מופיע תוכן לא תקף (כמות שלילית), יש להדפיס בקשה להודעת עדכון. כאשר בשני השדות קיים תוכן לא תקף, מודפסת הודעת שגיאה.

הפתרון בתרשים מלבני מוצג בציור 6-15 ובתרשים המבני - בציור 6-16. הפתרון בציור 6-17 מוצג בצורת טבלת החלטה. הקורא יכול

			צירופי תנאים					
		רשימת תנאים	1	2	3	4	5	6
אם ...	1							
(תנאי)	2							
	3							
	4							
		רשימת פעולות						
	1							
אז ...	2							
(פעולה)	3							
	4							
	5							

ציור 6-13 - מבנה של טבלת החלטה

טבלת החלטה מורכבת מארבעה חלקים:

בחלקה העליון של הטבלה נרשמים <u>התנאים</u>, בהם תלוי ביצוע או אי-ביצוע של פעילות מסוימת, ובחלק התחתון תירשמנה <u>הפעולות</u>, שיש לבצע. בחלק השמאלי העליון נרשמים התנאים, כל תנאי בשורה נפרדת. בחלק הימני העליון נרשמים <u>צירופי תנאים</u>, אשר מוצגים על ידי מספרים. ניתן לראות בחלק זה תשובות, אשר ניתנות לשאלות, שנשאלו בחלק השמאלי העליון, כמו למשל: "האם זהו סוף קובץ? אם כן, הדפס שורת סיכום".
משבצות ריקות מציינות תנאי או פעולה שאינם רלבנטיים.

כדי להמחיש את המבנה ואת השימוש בטבלת החלטה מוצגת דוגמה בציור 6-14, שבה מברר תלמיד בית-ספר לעצמו, מה ללבוש לפני צאתו לבית-הספר, בהתאם לתנאי מזג האויר.

			1	2	3	4
	1	יורד גשם, אך לא קר	כן	לא	לא	לא
אם ...	2	קר, אך בהיר		כן	לא	לא
(תנאי)	3	קר, מעונן			כן	לא
	1	ללבוש בגדים חמים		X	X	
	2	ללבוש בגדים רגילים				X
אז ...	3	ללבוש מעיל גשם	X			
(פעולה)	4	ללבוש רק סודר	X			
	5	לקחת מטרייה	X		X	

ציור 6-14 - טבלת החלטה לבעית "מה ללבוש"

את טור מס' 1 ניתן לקרוא כך:
<u>אם</u> יורד גשם, אך לא קר, <u>אז</u> לוקחים סודר, מעיל גשם ומטרייה.

את טור מס' 2 ניתן לקרוא כך:
<u>אם</u> לא יורד גשם, אך קר, <u>אז</u> יש ללבוש בגדים חמים בלבד.

ננסח עתה את הכללים למילוי טבלאות החלטה:

1. <u>רשימת התנאים</u> - התנאים ייכתבו בחלק השמאלי העליון של הטבלה. רצוי שכל שורה תכיל רק תנאי אחד כמו למשל "משכורת קטנה מ-280,000 שקל" או "מספר ילדים גדול מ-0".

2. <u>רשימת פעולות</u> - נכתבת בחלק השמאלי התחתון של הטבלה. רצוי למספר את הפעולות ולסדר אותן לפי סדר הגיוני, אך אין זה הכרחי. לעתים נדרשים לחזור על כתיבת פעולה מסוימת פעמים נוספות, כדי להבטיח בהירות ושלמות בקבוצת פעולות מסוימת.

3. <u>צירופי התנאים</u> - נרשמים בחלק הימני העליון של הטבלה, בעמודות ממוספרות. נוהגים לכתוב את המלים "כן", "לא" בשורות המתאימות לכל צירוף, אך יש ורושמים "1" כדי לסמן קיום תנאי מסוים ו-"0" לאי-קיום התנאי. בצירופים, שבהם לתנאי כלשהו אין משמעות, משאירים מרווח (BLANK), או רושמים את סימן המינוס.

4. <u>צירופי פעולות</u> - נכתבים או מסומנים בחלק הימני התחתון של הטבלה. נוהגים לסמן בסימן X את הפעולות, שיש לבצע, בהתאם לצירופי התנאים שמעליהם בטבלה. הצירוף הינו אוסף של פעולות אחדות, שיש לבצע, כאשר מתקיים צירוף של תנאים כלשהם.

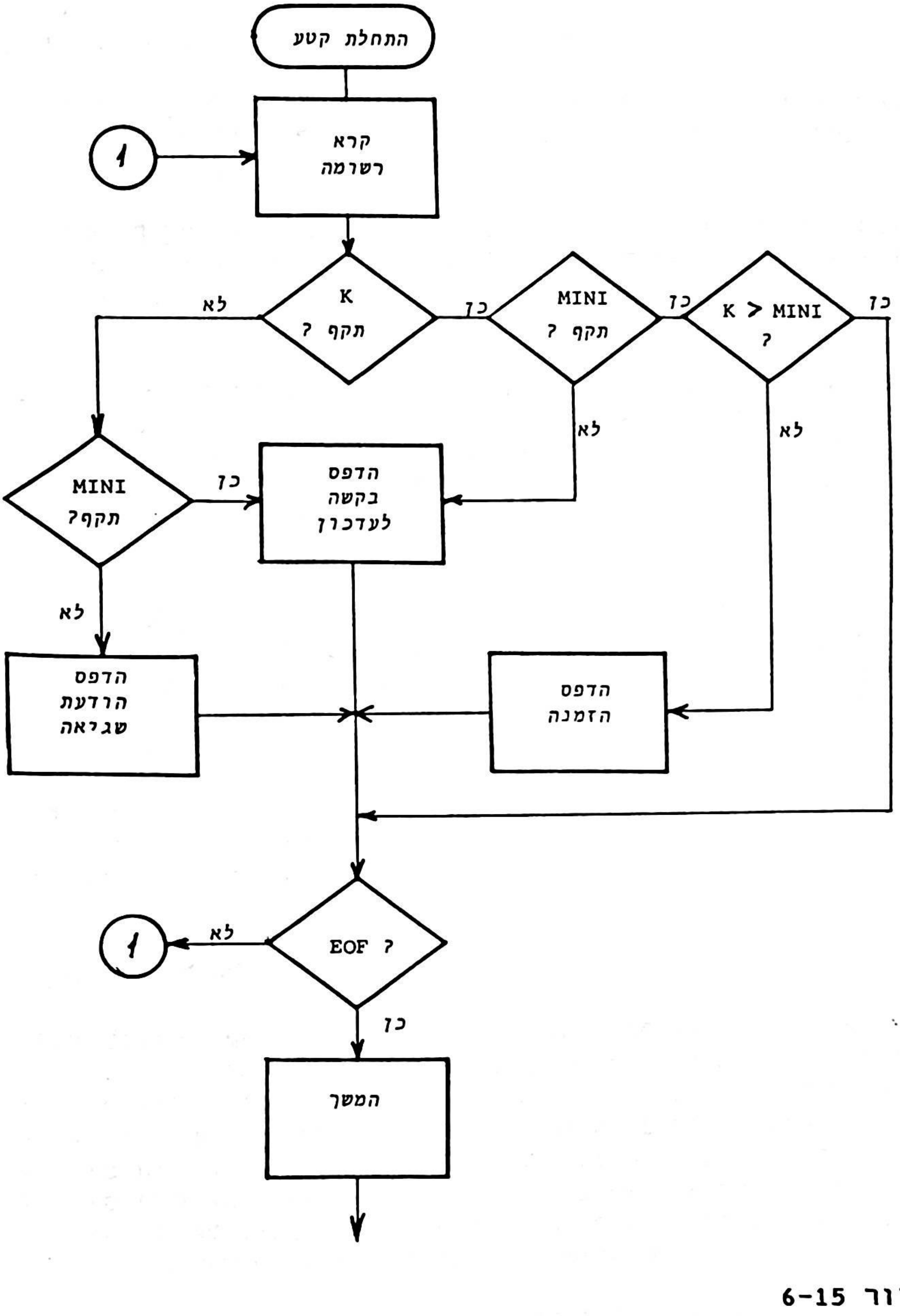
לשם סיכום, נציג עתה לדוגמה בעייה בניהול מחסן פריטים ופתרונה בשלש דרכים, בשיטת תרשים מלבני, שיטת תרשים מבני ובשיטת טבלת החלטה.

<u>בעיה בניהול מלאי</u>

ניקח לדוגמה קטע מתכנית גדולה יותר, אשר מטפלת בניהול מלאי במחסן. קטע זה חייב לבדוק, אם ברשומת הקלט מופיע תוכן תקף בשדה "כמות במחסן" ובשדה "כמות מינימום". אם תנאי זה מתקיים, יש לבדוק, אם הכמות במחסן נמוכה מן המינימום הדרוש, ואז יש להדפיס הזמנה. במקרה שהכמות במחסן גדולה מן המינימום הדרוש, עוברים לרשומה הבאה. במקרה שבאחד השדות לפחות מופיע תוכן לא תקף (כמות שלילית), יש להדפיס בקשה להודעת עדכון. כאשר בשני השדות קיים תוכן לא תקף, מודפסת הודעת שגיאה.

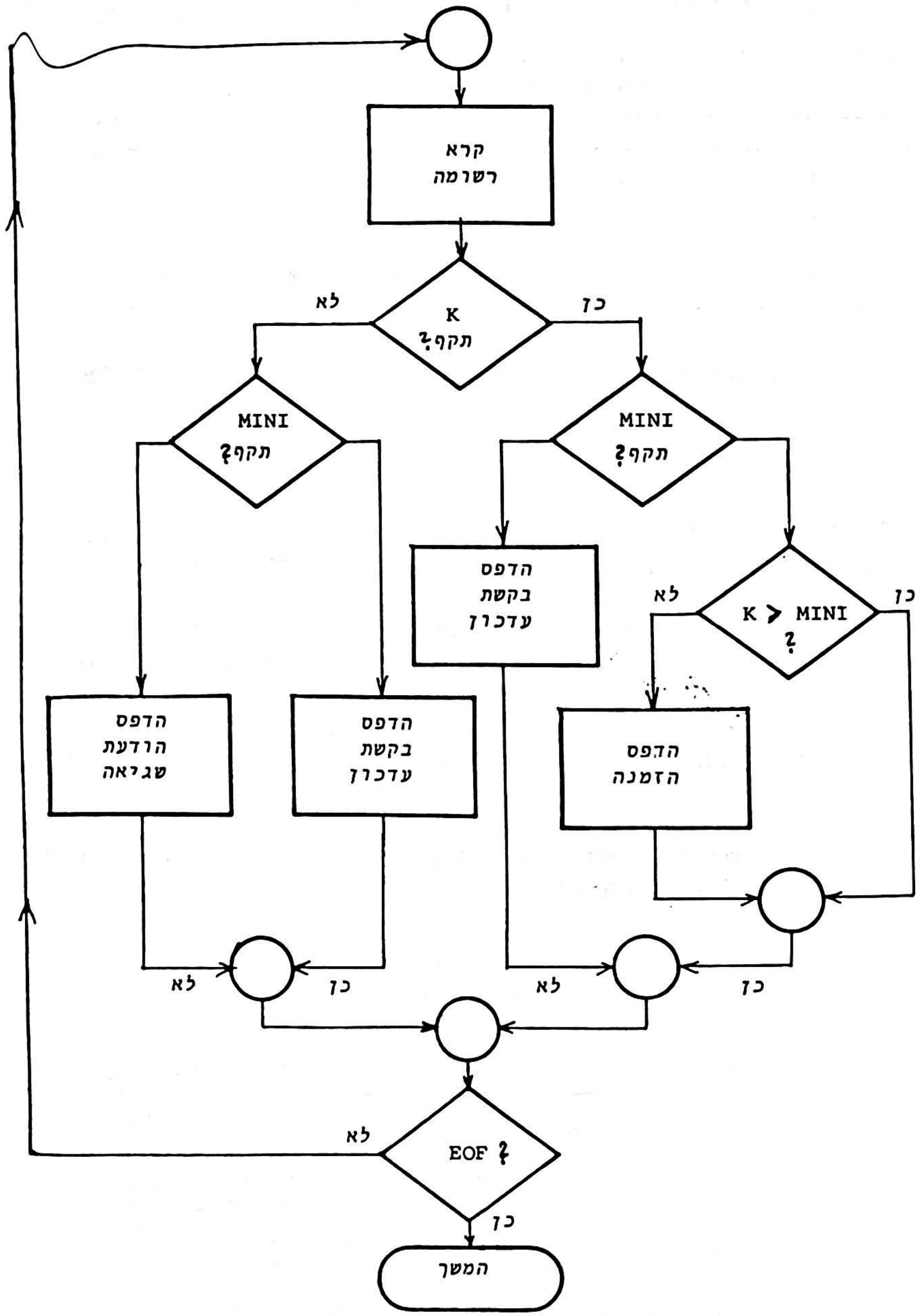
הפתרון בתרשים מלבני מוצג בציור 6-15 ובתרשים המבני - בציור 6-16. הפתרון בציור 6-17 מוצג בצורת טבלת החלטה. הקורא יכול

לשפוט בעצמו, שבמקרה זה למשל, הפתרון האחרון הינו קריא יותר
משני קודמיו.



ציור 6-15

- הפתרון מוצג בתרשים מלבני



ציור 6-16 - הפתרון מוצג בתרשים מבני

			1	2	3	4	5
	1	כמות במחסן - ערך תקף ?	כן	כן	כן	לא	לא
א ם	2	כמות מינימום - ערך תקף ?	כן	כן	לא	כן	לא
	3	מלאי גדול מכמות מינימום ?	כן	לא			
	1	הדפס שורה בדו"ח שגוים					X
א ז	2	הזמן פריט		X			
	3	הדפס בקשה לעדכון			X	X	
	4	עבור לרשומה הבאה	X	X	X	X	X

ציור 6-17 - פתרון לבעית המלאי בצורת טבלת החלטה

<u>הסבר לפתרון:</u>

בטור 1, בחלק העליון נבדקים שלושת התנאים, המוצגים בשאלות. <u>אם</u> שלושת התנאים מתקיימים, <u>אז</u> אפשר לקרוא את הרשומה הבאה.

בטור 2 מוצג המקרה בו הכמות במחסן נמוכה מן הכמות המינימלית. כאן יש להדפיס הזמנה לפריט ואחר-כך לעבור לקריאת הרשומה הבאה וכו'.

טורים 3 ו-4 מציינים שיש ערך לא תקף באחד השדות. מדפיסים בקשה לעדכון ועוברים לרשומה הבאה וכו'.

בטור 5 מוצג מצב שבו בשני השדות שבהם מטפלים יש ערך לא תקף. במקרה זה מדפיסים הודעת שגיאה, ועוברים גם כן לרשומה הבאה.



פרק 6 - שאלות

1. "החברה המטאורולוגית הישראלית" הציבה כלי מדידה למשקעים ולמידות החום בישובים רבים בארץ. תוצאות המדידה נאספות בסוף כל שבוע מהאנשים, האחראים לתחנות המדידה הללו, ומועלות על סרט מגנטי.

מבנה הרשומה בקובץ:

שם הישוב	תאריך	מידת החום	כמות משקעים

עליך להכין תרשים, אשר יקרא את הקובץ, וידפיס את הממוצע השנתי של מידות החום והמשקעים לאותו יישוב, בהנחות הבאות:

א. בהתחלת תניח, שכל הקובץ מתייחס ליישוב אחד.

ב. לאחר ביצוע אי תניח, שהקובץ מתייחס ל-300 יישובים בארץ.

ג. בכל יום נערכות מדידות, 360 פעם בשנה.

2. רובוט יושב על כיסא במרחק לא ידוע מקיר שלפניו. ברובוט מותקן מונה מאופס.
הרובוט נשמע להוראות, המופנות אליו, ולאחר כל הוראה הוא מבצע את הפעולה המתאימה בלבד.

ההוראות, שאפשר לתת לרובוט הן:

קום
שב
צעד פסיעה אחת קדימה
הסתובב
הוסף 1 למונה
החסר 1 מן המונה
בדוק : האם הקיר לפניך ?!
בדוק : האם המונה = 0 ?!

רשום תרשים של הוראות, שיש לתת לרובוט, כדי שהרובוט ייגש לקיר, ויחזור לשבת על הכיסא כקודם.

ציין בתרשים כל פעולה בתוך מלבן, וקשור את המלבנים באמצעות קו עם חץ, שיצביע על הפעולה הבאה לביצוע. את פעולות הבדיקה רשום בתוך מעוין.

3. הכן תרשים לפי כללי התכנות המבני לפתרון הבעיה הבאה:

משרד הרשוי חייב לשלוח רשיון נהיגה חדש לכל נהג, שקיבל רשיון כזה לפני שנתיים. כמו כן יש לשלוח רשיון נהיגה לכל האנשים, שעברו מבחני נהיגה בהצלחה ומלאו להם 18 שנים.

קובץ הזכאים נמצא על סרט מגנטי, ובו הרשומות המכילות, בין היתר, את השדות הבאים:

מספר ת.ז.	שם	כתובת	שנת לידה	מספר רשיון	תאריך אחרון של הרשיון

4. רשת חנויות "סופר-היפר" מפיצה כל חודש תקליטונים לכל החנויות, ועליהם מחירי היסוד של המצרכים, הנמכרים ע"י הרשת.
מבנה הרשומות:

מספר קטלוגי	שם פריט	מחיר יסודי	קוד מחלקה	מספר מחסן

עליך להכין תרשים לקריאת הרשומות והדפסת רשומת מצרכים בעלת מחירים הכוללים 15% מע"מ. אל תוסיף מע"מ למחירי הירקות, המסומנים בקוד 77.

5. בעיית הכדור הקל יותר:

א. הכן תרשים, שבעזרתו ניתן למצוא כדור מזוייף (קל יותר) מבין 9 כדורים, על-ידי שלוש שקילות בלבד במאזני שקילה.

ב. האם ניתן למצוא את הכדור המזוייף ע"י שתי שקילות?

6. הנהלה של בית-ספר מסוים מתכוונת להעניק פרסי לימודים לתלמידים המצטיינים שהציון הממוצע שלהם מעל 90. תלמידים, שהממוצע שלהם נע בין 80 ל-90, יצוינו לשבח ע"י הדפסת שמותיהם על לוחות המודעות.

הכן תרשים, אשר יבצע עיבוד זה כך, ששמות המצטיינים, שהציון שלהם מעל 90, ייכתבו על סרט מגנטי, ואם הציון מעל 80 - השמות יודפסו במדפסת.

פרק 7 - שפות תכנות

שפות התכנות מעניינות במיוחד. מצד אחד הן דומות לשפות "רגילות", שבני-אדם מכירים: ז"א הן מורכבות ממלים, משפטים, סימני פיסוק וכו', אולם מצד אחר לא נוכל למצוא עם או קבוצת אנשים, המדברים בשפת תכנות כלשהי (למרות העובדה, שלכמה שפות תכנות יש אפילו "ניבים" שונים...).

נזכיר עובדה נוספת: רב בני האדם לומדים בתחילה __לדבר__ בשפה מסוימת, אחרי-כן לומדים את כללי הכתיבה והתחביר של אותה שפה, ולבסוף כותבים בה. התהליך שונה לגבי שפות תכנות: בתחילה לומדים את רשימת הסימנים בהם מותר להשתמש, אחר-כך לומדים את כללי התחביר, ולבסוף - אין מדברים בהן כלל...

שפות התכנות הינן המצאה מודרנית למדי. שפת התכנות הראשונה "נולדה" סמוך לאמצע המאה העשרים, ומאז פותחו עשרות ואולי מאות שפות. למרות מספרן הגדול יחסית - פיגרו שפות התכנות בהתפתחותן אחרי ההתפתחות המסחררת של מאות סוגים של ציוד לעיבוד הנתונים. אחד מהנשיאים של חברת CDC, ויליאם נוריס, אמר: "הדברים הקלים בשטח המחשב בוצעו זה מכבר. היום פשוט הרבה יותר לפתח ציוד למחשבים, מאשר לכתוב תכניות בעבורם... "

בפרק זה נסקור את ההתפתחות של שפות התכנות, ונציג בקצרה מספר שפות ידועות, המקובלות בעולם כולו. פרק מיוחד יוקדש לשפת BASIC, אשר נעשתה פופולרית בשנים האחרונות, בעידן המחשבים הביתיים והמחשבים האישיים.

7.1 תכנות בשפת מכונה

מחשב יכול לבצע הוראות, המוצגות לו בשפה שהוא מבין - בשפה הבינרית. לכן, בדור הראשון של המחשבים היתה מלאכת התכנות קשה ומייגעת. בעת כתיבה של תכנית מותר היה להשתמש בשני סימנים בלבד: 1, 0 ולכן הוראה __בשפת מכונה__ (MACHINE LANGUAGE) עשויה היתה להראות כך:

101000111011000010010000111100

כלומר, הוראה בשפת מכונה היתה סדרה של סימנים של "הדלקה" (1) או של "כיבוי" (0) של מעגלים אלקטרוניים מסוימים. שיטת כתיבה זו היתה בלתי יעילה, ודרשה זמן רב לתכנות. קל מאד היה לטעות

בכתיבה, כי כמעט בלתי אפשרי היה להבחין, אם משמיטים ספרה כלשהי של אפס, או מוסיפים בטעות ספרה של 1 במקום כלשהו. חיסרון נוסף של השיטה נובע מהעובדה שכל אדם, שכתב תכניות בעבור מחשב מסוים היה נדרש להכיר היטב את המבנה הפנימי שלו ולהקפיד לרשום את הכתובת המדויקת של הנתונים בזיכרון.

נמחיש זאת בדוגמה:

נניח שהיה צריך לחבר שני מספרים. כדי שהמחשב יוכל לחברם, הם היו חייבים להימצא בזיכרון. התכניתן היה צריך לדעת את הכתובת המדויקת של שני השדות (קבוצת הבתים בזיכרון) שהכילו את המספרים. כמו כן היה ידוע, שניתן לחבר מספרים בתוך <u>אוגר</u> (REGISTER) בלבד.

לכן הוראות המחשב נראו כך:

כתובת המחובר	מספר אוגר	פעולה

לפיכך, פעולת חיבור של שני מספרים בשפת מכונה נכתבה למעשה ע"י שלוש הוראות, כמו למשל:

1. טעינה של המחובר הראשון לתוך אוגר מסוים

   01011000011000000000110000111100

2. חיבור שני המחוברים

   01011010011000000000001100010110

3. אחסון התוצאה של החיבור בתא מסוים בזיכרון

   01101100010100000000110100101110

הקשיים בכתיבת תכניות בשפת מכונה חייבו חיפוש דרכים שונות, אשר תקלנה, ולו במקצת, על מלאכת הכתיבה.

7.2 תכנות בשפה סמלית

המאמצים למציאת שיטות תכנות נוחות יותר נשאו פרי בראשית שנות החמישים. אז הוצגה בארה"ב <u>שפת האסמבלר</u> (ASSEMBLER LANGUAGE). בשפה זו מותר היה להשתמש בכל 26

האותיות של ה-ABC, בכל הספרות מ-0 עד 9 וכן בכמה סימני פיסוק. הוראת חיבור של שני מספרים בשפה זו נראתה כך:

```
(LOAD)   L    6,X
(ADD)    A    6,Y
(STORE)  ST   6,Z
```

האותיות משמאל הן פקודות, שמוצגות בצורה סימלית. הספרה מציינת מספר אוגר, בו נעשה החישוב, והאות בצד ימין בכל הוראה מסמלת תא זיכרון (סמלי, לא מוגדר). ההבדל היה גדול. אפשרות השימוש בסמלים השונים הקלה מאד על התכניתנים, מכיון שהם היו שאולים מהתווים, המשמשים בכתיבה יומיומית. נוצר המונח: <u>שפה סימלית</u> (SYMBOLIC LANGUAGE) והכוונה - כתיבה בשפת תכנות, שאינה שפה בינרית אלא שפת סמלים.

יש לשים לב, שההוראות בשפה הסמלית נכתבות לפי אופן הכתיבה של שפת מכונה. כל הוראה שומרת על תבנית של הוראת מכונה, ובמקום כל הוראת מכונה נכתבת הוראה בשפה הסמלית. שפות כאלה נקראות <u>שפות סף.</u> דוגמה טובה לשפה כזו היא שפת ה-ASSEMBLER של חברת יבמ.

שפה זו נשארה נאמנה בתפישתה לשפת מכונה, ולכן היא נקראת גם <u>שפה ייעודת מכונה</u> (MACHINE ORIENTED LANGUAGE), או <u>שפה ברמה נמוכה</u> (LOW LEVEL LANGUAGE). נאמנות זו מתבטאת בכך שקיימת התאמה חד-ערכית בין הוראה בשפת המכונה לבין הוראה בשפת האסמבלר. הרי לדוגמה פקודות אחדות:

שפת מכונה	אסמבלר	
01011000	L	(LOAD)
01011010	A	(ADD)
01011011	ST	(STORE)

יש לציין, שלכל סוג של מחשב קיימת שפת אסמבלר שונה. זה גם מובן לאור הקביעה, שיש הבדל בין המחשבים השונים, ושפת האסמבלר הוגדרה כשפה ייעודת מכונה. בציור 1-7 מוצגת, לדוגמה, תכנית קצרה בשפת האסמבלר של מערכת 370 של חברת יבמ.

כתיבת תכנית מתבצעת על טופסי קידוד מיוחדים, אשר מחולקים לשלושה חלקים, כפי שמוצג בציור: חלק 1, מטור 1 עד 8, מיועד למתן שמות סמליים לכתובות באחסנה. הקטע בין טורים 10-15 מיועד לכתיבת הפעולה, כגון חבר, טען, או הגדר שדה באחסנה (DS). מטור 16 והלאה כותבים את השמות של <u>נושאי הפעולה</u> (OPERANDS). אלה למעשה השמות הסמליים לאותם שדות, שהוגדרו בעזרת הוראות הגדרה בתחתית הציור (ראה לדוגמה RESULT, פעם בהגדרה - DS RESULT, ופעם בנושא פעולה ST 2,RESULT).

IBM

IBM System 360 Assembler Coding Form

PROGA

J.J. JONES

```
PROGA    START  0
BEGIN    BALR   11,0
         USING  *,11
         L      2,DATA          LOAD REGISTER 2
         A      2,CON           ADD 10
         SLA    2,1             THIS HAS EFFECT OF MULTIPLYING BY 2
         S      2,DATA+4        NOTE RELATIVE ADDRESSING
         ST     2,RESULT
         L      6,BIN1          BEGIN SECOND PROBLEM
         A      6,BIN2
         CVD    6,DEC           CONVERT TO DECIMAL
         EOJ                    END OF JOB
DATA     DC     F'25'
         DC     F'15'
CON      DC     F'10'
RESULT   DS     F
BIN1     DC     F'12'
BIN2     DC     F'78'
DEC      DS     D
         END    BEGIN
```

ציור 7-1 - תכנית בשפת אסמבלר של יבמ

התכנית טוענת את המספר 25 (DATA) לאוגר 2, אח"כ מוסיפה לו את תוכן שדה CON(10), מכפילה את הסכום פי 2 ומאחסנת את התוצאה בשדה RESULT. לאחר מכן נטען המספר 12 מ-BIN1 לאוגר 6, מתוסף לו המספר 18 משדה BIN2, והתוצאה מוסבת למספר עשרוני, ונשמרת בשדה DEC.

התכנית עוברת, כמובן, תרגום לשפת מחשב. הפלט של ה-ASSEMBLER מוצג בציור 7-2. בחלקו הימני של הציור, מודפסת התכנית, שהוצגה בציור 7-1, ובחלקו השמאלי - התכנית בשפת מכונה. התכנית מוצגת בשפה הקסה-דצימלית במקום בשפה הבינרית (ראה נספח א').

```
                                                                              PAGE   1
  LOC     OBJECT CODE    ADDR1   ADDR2   STMT   SOURCE STATEMENT             08/08/80

000100                                     1   PROGA    START  256
000100    05B0                             2   BEGIN    BALR   11.0
000102                                     3            USING  *.11
                                           4   * THIS PROGRAM COMPUTES TWO VALUES
000102    5820   B02A            0012C     5            L      2.DATA      LOAD REGISTER 2
000106    5A20   B032            00134     6            A      2.CON       ADD 10
00010A    8B20   0001            00001     7            SLA    2.1         THIS HAS EFFECT OF MULTIPLYING BY 2
00010E    5B20   B02E            00130     8            S      2.DATA+4    NOTE RELATIVE ADDRESSING
000112    5020   B036            00138     9            ST     2.RESULT
000116    5860   B03A            0013C    10            L      6.BIN1      BEGIN SECOND PROBLEM
00011A    5A60   B03E            00140    11            A      6.BIN2
00011E    4E60   B046            00148    12            CVD    6.DEC       CONVERT TO DECIMAL
                                          13            EOJ                END OF JOB
00012C    00000019                        16   DATA     DC     F'25'       F RESERVES ONE FULL WORD
000130    0000000F                        17            DC     F'15'
000134    0000000A                        18   CON      DC     F'10'
000138                                    19   RESULT   DS     F
00013C    0000000C                        20   BIN1     DC     F'12'
000140    0000004E                        21   BIN2     DC     F'78'
000148                                    22   DEC      DS     D           D RESERVES ONE DOUBLEWORD
000100                                    23            END    BEGIN
```

ציור 7-2 - פלט ה-ASSEMBLER

לסיכום, התכנות בשפת סף הביא אמנם הקלה לתכניתנים, כי אפשר היה להשתמש בסמלים מוכרים ומקובלים על הכל, אולם חיש מהר נתגלו חסרונות:

1. לכל מחשב שפת סף משלו, וככל שרבו המחשבים התרבו השפות;

2. כל תכניתן חייב היה ללמוד היטב את המחשב עליו הוא עבד, לימוד כזה דרש זמן רב;

3. תוספת או השמטת הוראה אחת בלבד השפיעו לפעמים על התכנית כולה, והיה צורך לשכתב חלק גדול מחדש.

לכן נמשכו המאמצים להקל על מלאכת התכנות, אך על כך - בסעיף הבא.

7.3 שפות תכנות עיליות

באמצע שנת 1982 התאספו בארה"ב 13 גברים ונשים בעלי שערות שיבה כדי לחגוג את יום ההולדת ה-25 של שפת ה-FORTRAN. האם הולדת שפת הפורטרן היא מאורע כה חשוב, שיש לחגוג אותו?

מסתבר שכן. באמצע שנות ה-50 התארגנה קבוצת אנשים בחברת יבמ במטרה ליצור שפת תכנות, אשר מצד אחד תהיה נוחה לשימוש, ומצד אחר לא תדרוש הבנה מרובה במעגלים הפנימיים של מערכת המחשב. אותם אנשים בעלי היזמה היו ברובם מהנדסים ומתמטיקאים, אשר נזקקו למחשב ככלי חישוב מהיר. הם לא רצו "לבזבז" את זמנם היקר בכתיבת תכניות בשפת מכונה או בשפת הסף, אלא רצו לפתור בעזרת המחשב בעיות שונות במסגרת עבודתם.

כתוצאה מן המאמצים שלהם נוצרה שפת הפורטרן, שפת התכנות הראשונה אשר סימלה את עידן <u>השפות ייעודות הבעיה</u> או <u>שפות יישומיות</u> (PROBLEM ORIENTED LANGUAGES). בעקבות שפת הפורטרן "נולדו" שפת הקובול (COBOL) ואח"כ RPG, PL/I, ALGOL, BASIC, PASCAL ואחרות. מה מייחד את השפות הללו?

1. הן נכתבות באותם סמלים, אשר משמשים אותנו בשפה האנגלית (ABC...) או במתמטיקה (+,-,= ועוד).

2. הן דומות במבנה שלהן למשפטים של השפה האנגלית או לשפת המתמטיקה.

3. הן חסכוניות בכתיבה, כי ניתן לכתוב תכנית בפחות הוראות מאשר בשפת מכונה (הודות למהדירים - COMPILERS). למשל, חיבור שני מספרים A,B, שנכתב בשלוש שורות בשפת מכונה או בשפת סף, נכתב ב-COBOL בשורה אחת: ADD A TO B.

4. הן נכתבות בצורה חפשית הרבה יותר (FREE FORM), מאשר שפת מכונה או שפת סף.

5. הן בלתי תלויות (כמעט) בסוג המחשב.

כאן דרוש הסבר מפורט יותר: בשפת מכונה או בשפת סף נכתבו התכניות בתוך התחשבות מירבית בסוג המחשב, בו הן נתבצעו. התכניתן חייב היה להכיר את המחשב ואת תכונותיו המיוחדות, כדי שיוכל לתת לו את הוראות התכנית. בשפת התכנות העילית, מבנה השפה וחוקיה אינם דורשים ידע מפורט של מבנה המחשב ותכונותיו. בשפות הללו ההוראות מנוסחות כך, שהן תואמות כמעט כל מחשב, שבו ניתן להזין מהדיר אשר מתאים לשפה מסוימת.

התכונה של <u>אי-תלות במחשב מסוים</u> (MACHINE INDEPENDENCE) חשובה. היא מאפשרת לעבור מסוג מחשב אחד לסוג מחשב אחר, ואפילו הם לא של אותו יצרן. תכונה זו תורמת לייעול העבודה ולניצול נכון של משאבי כח-אדם יקרים, כי התכניתנים עוסקים בפתרון הבעיה בלבד בלי לטפל בעניינים, הנוגעים לצורת פעילות המחשב. כאן המקום לציין, שהמצב בפועל איננו מזהיר כל כך, כי קיימת תלות מסוימת, בעיקר בתחום של ארגון הנתונים המאוחסנים ביחידות האחסנה של המחשב. יצרנים שונים הרחיבו את השפות הסטנדרטיות, ובכך הכניסו שוני רב יותר בין השפות.

להלן מספר דוגמאות של קטעי תכנית, שנכתבו כולן בשפות עיליות:

COBOL

```
READ SALARY-FILE RECORD AT END GOTO FINISH.
COMPUTE GROSS-SALARY = HOURS MULT BY PAY1.
```

PL/I

```
IF PARTNO = 9999 GOTO TOTAL
ELSE GOTO READ RECORD
```

FORTRAN

```
READ A,B,C
X1=(-B+(B**2 - 4*A*C)**0.5)/(2*A)
PRINT X1
```

הדוגמאות הללו מראות, שלכל הקטעים דמיון לשפה האנגלית או לנוסחה מתמטית. בגלל הדמיון הזה וכן בגלל התכונות, שנמנו לעיל, נקראות השפות הללו <u>שפות עיליות</u> (HIGH LEVEL LANGUAGES), בניגוד לשפות הסף, אשר בגלל הדמיון הרב של מבנה ההוראות שלהן לשפת המכונה, הן נקראות <u>שפות ברמה נמוכה</u> (LOW-LEVEL LANGUAGES).

הבה נסכם את ההבדלים בין שפות עיליות לבין שפות נמוכות:

1. שפות עיליות משחררות את המתכנת מתלות בתכונות המיוחדות של מחשב מסוים, כגון אוגרים, הצגה פנימית של נתונים, ערוצים לקלט/פלט ועוד;

2. תכניות בשפת מחשב אפשר להריץ לא רק על סוג מחשב אחד אלא על סוגים שונים, בלא להכנים שינוי בתכנית או בתוך שינויים קלים בלבד, כל עוד משתמשים בשפה הסטנדרטית;

3. תכניות בשפה עילית נכתבות בצורה חופשית ובשפה הדומה לשפה המדוברת או לשפת המתמטיקה;

4. תכניות בשפות עיליות הן בעלות זיקה אל היישום עצמו;

5. שפה עילית מאפשרת לכתוב תכניות בקצב מהיר יותר מאשר שפות מכונה.

בהמשך נציג בקצרה את שפות התכנות המקובלות ביותר.

7.4 פורטרן (FORTRAN)

שפת הפורטרן היא שפת התכנות העילית הוותיקה ביותר. כאמור, היא פותחה באמצע שנות ה-50, והתכניות הראשונות הורצו בהצלחה ב-1957. שם השפה נגזר משתי מלים אנגליות <u>FOR</u>MULA <u>TRAN</u>SLATOR (בעברית: מתרגם נוסחאות). השפה פותחה בעיקר על ידי מהנדסים ומתמטיקאים, שהיו זקוקים באותה עת לשיטה נוחה, פשוטה ומהירה של מתן הוראות למחשב, כאשר רוב ההוראות היו קשורות בנוסחאות שונות.

<u>תכונות השפה</u>

האלמנט היסודי של שפת הפורטרן הוא <u>ההוראה</u> (STATEMENT). סוגי הוראה ודוגמאות להם מוצגים להלן:

1. <u>הוראת השמה (ASSIGNMENT STATEMENT)</u>

הצורה הכללית של ההוראה :

ביטוי אריתמטי = שם של משתנה בודד

דוגמה להוראת השמה :

```
AVERG = TOTAL/COUNT1
```

פרוש ההוראה למחשב : "חשב את ערכו של הביטוי, המופיע מימין לסימן השוויון (במקרה זה חלק את תוכן שדה TOTAL בתוכן שדה COUNT1), ושים את התוצאה בשדה בשם AVERG".

2. <u>הוראות קלט/פלט (INPUT/OUTPUT STATEMENTS) והוראת תבנית (FORMAT) השייכת להן</u>

הצורה הכללית של הוראת קלט:

```
      READ M  שם-שדה, שם-שדה
   M  FORMAT (תבנית שדה, תבנית שדה)
```

דוגמה להוראת קלט:

```
      READ 20 KAMUT,PRICE,IDAY
  20  FORMAT(I4,F5.2,I2)
```

תיאור ההוראה:

ההוראה מורכבת משני חלקים: החלק הראשון מודיע למחשב, מה לקרוא, והחלק השני - איך לקרוא.

בדוגמה לעיל החלק הראשון מורה למחשב לקרוא את הנתונים לתוך שדות KAMUT, PRICE ו-IDAY (לפי הסדר הזה) כאשר:

- KAMUT הינו מספר שלם בן ארבע ספרות,
- PRICE הינו מספר במבנה של שבר עשרוני בעל 5 ספרות ומתוכן 2 ספרות הן מימין לנקודה העשרונית.
- IDAY מספר שלם בן 2 ספרות.

הערה:
הכללים למתן שמות לשדות חורגים מדיוננו כאן. נסתפק רק בכך, שנאמר כי מותר לתת שם שאורכו אינו עולה על 6 תווים, והוא חייב להתחיל באות.

הוראת כתיבה דומה להוראת קריאה:

```
      WRITE 8,TOTAL
   8  FORMAT(F6.2)
```

פירוש ההוראה:

כתוב תוכן שדה TOTAL ליחידת פלט. השדה מתואר בהוראת FORMAT מספר 8, אשר מציינת ש-TOTAL הינו מספר בן 6 ספרות ומתוכן 2 ספרות הן מימין לנקודה העשרונית.

יש לשים לב, שהמספרים הצמודים למלים READ ,WRITE הם מספרים של <u>הוראות תבנית</u> (FORMAT), השייכות להוראת קלט או פלט. למעשה כל הוראה בשפת פורטרן יכולה לקבל מספר כלשהו, בתנאי שאין שתי הוראות בעלות מספר זהה.

3. <u>הוראת בקרה (CONTROL STATEMENT)</u>

ההוראות בתכנית הפורטרן מתבצעות לפי הסדר בו הן רשומות, אולם לפעמים יש צורך לבצע הוראה, שאינה ברצף הרגיל של ההוראות, או להפסיק את ביצוע התכנית. הוראות כאלה נקראות <u>הוראות בקרה:</u>

STOP - הינה הוראה לעצירת הביצוע של התכנית.
GOTO - הינה הוראת הסתעפות. דוגמה: GOTO 250.

פירוש הוראה GOTO: בנקודה זו אין להמשיך בביצוע השוטף של ההוראות, אלא להסתעף להוראה מס' 250, ולהמשיך לבצע משם והלאה. תנאי הכרחי הוא שתימצא בתכנית הוראה אחת, ואחת בלבד, שמספרה 250.

4. <u>הוראת התניה והסתעפות (הוראת IF)</u>

צורה כללית:

IF (ביטוי) הוראת הסתעפות

דוגמה להוראת IF : IF (A.EQ.B) GO TO 300

פירוש הוראה זו: אם A שווה B, הסתעף אל הוראה מספר 300. אם התנאי A=B אינו מתקיים, ההסתעפות לא תתבצע, והביצוע ימשיך בהוראה, שכתובה לאחר הוראת IF.

קיימים סוגי הוראות נוספים בשפת הפורטרן. תיאורים והסברים עליהם חורגים ממסגרת הדיון זה. אולם מן הראוי להציג כאן תכנית קצרה בפורטרן ולהסביר אותה לקורא, הוראה אחר הוראה.

```
      PROGRAM TEMP (INPUT,OUTPUT)
      IDAYS = 0
    1 READ 10,CELS
   10 FORMAT(F4.2)
      FAHREN = (9*CELS)/5+32
      PRINT 20,FAHREN
   20 FORMAT(*0*,F5.2)
      IDAYS=IDAYS+1
      IF (IDAYS.LT.360) GOTO 1
      STOP
      END
```

הסבר לתכנית TEMP בשפת פורטרן

תכנית זו קוראת 360 רשומות של קובץ קלט, בו נרשמו בישוב מסוים, במשך שנה, מידות חום יומיות במעלות צלסיוס. מטרת התכנית לקרוא את הקובץ, להפוך כל מידת חום מצלסיוס לפרנהייט ולהדפיס אותה.

שורה 1 זוהי התחלת התכנית, מעין שורת פתיחה. היא מכריזה על שם התכנית (TEMP), ומודיעה על שימוש באמצעי קלט ובאמצעי פלט. שורה זו אינה מופיעה בניבים מסוימים של השפה.

שורה 2 איפוס המונה של מספר הימים.

שורה 3 קריאת רשומה מקובץ הקלט.

שורה 4 הוראת תבנית המודיעה, שמידת החום ברשומת הקלט נרשמה בארבע ספרות, מהן שתיים מימין לנקודה העשרונית (למשל 23.17).

שורה 5 הוראת השמה. בתחילה יחושב הביטוי מימין לסימן "=", ואחר-כך תועבר תוצאת החישוב לשדה בשם FAHREN.

שורה 6 הדפסה של תוכן שדה FAHREN, שחושב זה עתה.

שורה 7 הוראת תבנית, המורה להדפיס את תוכן FAHREN בשורה חדשה בכל פעם.

שורה 8 קידום מונה הימים באחד.

שורה 9 הביטוי בתוך הסוגריים בודק אם מספרי הימים (תוכן מונה IDAYS) קטן מ-360. אם תנאי זה מתקיים, יבצע המחשב הסתעפות להוראת קריאה של רשומה חדשה, שמספרה 1.

שורה 10 זוהי הוראה לעצירת הביצוע של התכנית. המחשב יבצע הוראה זו רק באותו מקרה שהתנאי הכתוב בשורה 9 אינו מתקיים, ז.א. מונה הימים הגיע ל-360.

שורה 11 הוראת סיום התכנית.

השם COBOL ‏(COMMON BUSINESS ORIENTED LANGUAGE) מורכב מחמש אותיות של ראשי התיבות המסומנות, והוא רומז על כך, שיוצרי השפה התכוונו לפתח שפת תכנות, אשר תתאים לחיי המסחר היומיומיים.

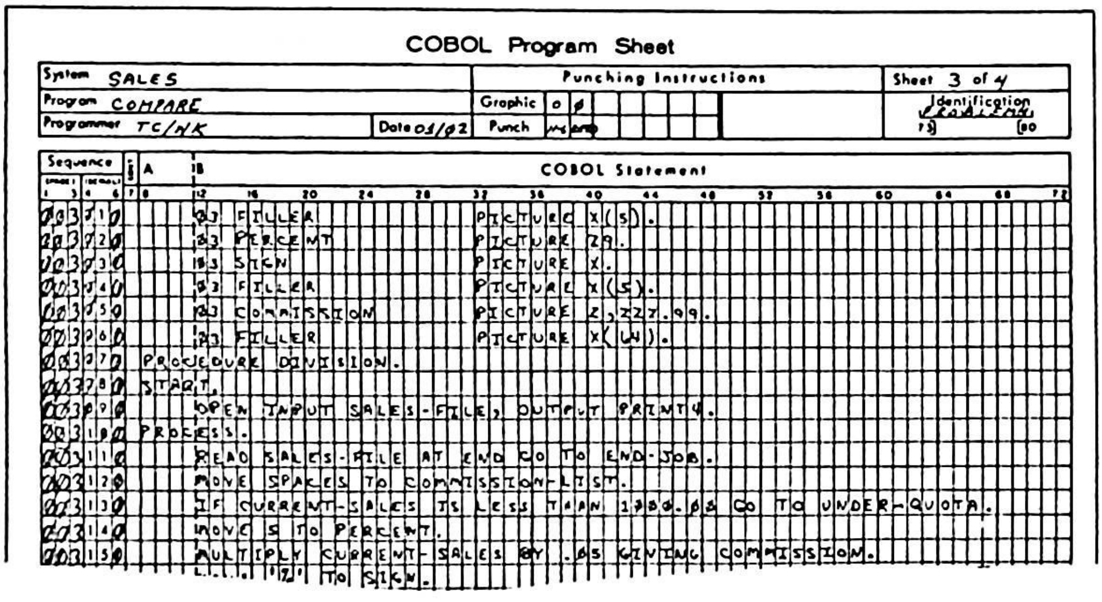
השנה היתה שנת 1959, ומפתחי שפת הקובול היו חברי ועדה, שאת הקמתה יזמו אנשי המסחר והממשלה בארה"ב. המטרה המוצהרת היתה ליצור שפת תכנות, שתכונותיה העיקריות תהיינה פשטות בכתיבה וחסכון בזמן הכתיבה. היות שאין דבר קל בעיני האמריקנים מאשר לכתוב באנגלית, הם תכננו ליצור שפת תכנות "אנגלית". ואמנם שפת הקובול דומה מאד לשפה האנגלית. קטעים שלימים של תכנית בשפת קובול מובנים למי שקורא אנגלית.

```
READ-AND-CHECK-RECORD.
     READ BILLING-FILE AT END GOTO END-OF-RUN.
     IF NUMBER-OF-TRANSACTIONS IS EQUAL TO ZERO, OR
     CURRENT-BALANCE IS NEGATIVE,
     GOTO READ-AND-CHECK-RECORD.
 CONT.
```

ציור 7-3 - קטע מתכנית קובול

הסבר לציור 7-3

קטע זה מתחיל בשם: READ-AND-CHECK-RECORD. זהו שם הפסקה כולה. בפסקה ישנה הוראת קריאה לקובץ הקלט (BILLING-FILE). אם בתוך כדי קריאה מגיעים לסוף הקובץ, יש להסתעף לפסקה בשם END-OF-RUN, בכל מקרה אחר ממשיכים בביצוע. הביצוע כולל את הבדיקה של שני שדות. אם שדה של מספר העסקות שווה לאפס או המאזן הנוכחי שלילי, יש לחזור להתחלת הפסקה ולקרוא רשומה חדשה. בכל מקרה אחר יש להמשיך בפסקה הבאה בשם CONT.


COBOL Program Sheet

System SALES — Punching Instructions — Sheet 3 of 4
Program COMPARE — Graphic — Identification
Programmer TC/NK — Date 01/02 — Punch

```
003010    03 FILLER              PICTURE X(5).
003020    03 PERCENT             PICTURE Z9.
003030    03 SIGN                PICTURE X.
003040    03 FILLER              PICTURE X(5).
003050    03 COMMISSION          PICTURE Z,ZZZ.99.
003060    03 FILLER              PICTURE X(4).
003070 PROCEDURE DIVISION.
003080 START.
003090    OPEN INPUT SALES-FILE, OUTPUT PRINT4.
003100 PROCESS.
003110    READ SALES-FILE AT END GO TO END-JOB.
003120    MOVE SPACES TO COMMISSION-LIST.
003130    IF CURRENT-SALES IS LESS THAN 1750.00 GO TO UNDER-QUOTA.
003140    MOVE 5 TO PERCENT.
003150    MULTIPLY CURRENT-SALES BY .05 GIVING COMMISSION.
             '%' TO SIGN.
```

תכונה נוספת של תכנות בקובול הוא הדמיון למבנה של ספר. כמו שספר מחולק לחלקים, לפרקים, לפסקאות ולמשפטים, כך גם התכנית בקובול.

האלמנט הבסיסי בשפה הינו <u>המשפט</u> (SENTENCE) או הוראה, בדומה לפורטרן. הרמה הבאה הינה <u>הפסקה</u> (PARAGRAPH). אח"כ באים <u>הפרקים</u> (SECTIONS) ולבסוף - <u>החטיבות</u> (DIVISIONS).

<u>המרכיבים היסודיים של שפת הקובול</u>

1. <u>סימנים</u> המותרים לשימוש בקובול:

- האותיות של ABC אנגלי
- הספרות 0-9
- סימני הכתיבה הבאים: =,X,-,',;, ,<,/,*,(,),,(פסיק) וסימן למרווח בין המלים.

2. <u>מלים שמורות</u> (RESERVED WORDS)

בשפת הקובול ישנה קבוצה של מלים אנגליות בעלות משמעות מיוחדת, שהשימוש בהן מותר רק בהקשר מסוים וכפוף לכללים מסוימים. רשימה שלימה של מלים אלה וכללי השימוש בהן נמצאים בכל ספר תכנות בקובול. מספר המלים השמורות מתקרב ל-100. להלן כמה מלים שמורות לדוגמה:

ADD	END
AREA	GO
BLOCK	READ
CONSOLE	IF
DISPLAY	ZERO

3. <u>תכנות בקובול</u>

תכנית קובול נכתבת בטופסי קידוד מיוחדים, המחולקים ל-4 עמודות ולשורות אופקיות, כשבכל אחת 80 טורים.

<u>חלקי הטופס:</u>

א. חלק שמאלי (טורים 6 - 1) משמש למספור ההוראה,

ב. חלק A (טורים 11- 8) משמש בעיקר לציון סמלים, כמו כתובת סמלית, שם פרק או פסקה.

ג. חלק B (טורים 12-72) משמש לכתיבת הוראות התכנית.

ד. חלק ימני (טורים 73-80) משמש לזיהוי התכנית.

מבנה התכנית בקובול

תכנית קובול מורכבת מארבע חטיבות (DIVISIONS) עיקריות:

1. IDENTIFICATION DIVISION - חטיבת הזיהוי

חטיבה זו מהווה את החלק הראשון של כל תכנית קובול. בחלק זה נכתבים פרטים מזהים של התכנית והתכניתן (שם תכנית, שם תכניתן, תאריך כתיבה ועוד).

2. ENVIRONMENT DIVISION - חטיבת הסביבה

חטיבה זו מכילה את התיאור של המחשב, בו "תרוץ" התכנית, וציון אמצעי קלט/פלט, שיופעלו במהלך ביצוע התכנית.

הערה: נאמר לעיל, שתכניות בשפות העיליות הינן בלתי תלויות בסוג המחשב, שבו הן פועלות. קביעה זו נכונה גם כאן, בתנאי שחטיבה זו תותאם כל פעם לסביבה (ENVIRONMENT), שבה תופעל התכנית. עם זאת ייתכן שיהיה צורך לערוך שינויים כלשהם גם בתכנית עצמה.

3. DATA DIVISION - חטיבת הנתונים

חטיבה זו מתארת את פרטי הנתונים, שיעובדו על ידי התכנית. התיאור יכלול פרטים על קובצי הקלט והפלט, על שטח עבודה בזיכרון המחשב, כגון מונים, כותרות לדו"חות וקבועים (CONSTANTS) אחרים.

4. PROCEDURE DIVISION - חטיבת הנהלים

החטיבה האחרונה מכילה למעשה את הוראות התכנית, כפי שהכרנו אותן בפרקים קודמים. כאן נכתבות כל ההוראות, שעל המחשב לבצע. בסיום כל הוראה יש לרשום נקודה.

להלן תכנית קצרה בקובול:

יש לקרוא כרטיס אחד מקוראת הכרטיסים, להוסיף את תאריך הביצוע ולהדפיסו. התאריך יסופק על ידי מפעיל המחשב. מבנה הפלט הוא:

PRINT DATE
XX/XX/XX ___ (תוכן הכרטים 80 טורים)
(תאריך) (3 רווחים)

<u>התכנית:</u>

```
      A    B
 1    IDENTIFICATION DIVISION.
 2    PROGRAM-ID. 'TARGIL.
 3    DATA DIVISION.
 4    WORKING-STORAGE SECTION.
 5    77 MESSAGE PIC X(10) VALUE 'PRINT DATE'.
 6    01 KARTIS-DATE.
 7        03 KARTIS PIC X(80).
 8        03 FILLER PIC X(3) VALUE SPACE.
 9        03 DATE PIC X(8).
10    PROCEDURE DIVISION.
11    KETA-1.
12        ACCEPT KARTIS.
13        DISPLAY MESSAGE UPON CONSOLE.
14        ACCEPT DATE FROM CONSOLE.
16        DISPLAY KARTIS-DATE.
17        STOP RUN.
```

<u>הסבר:</u>

א. שים לב למבנה התכנית הכולל:

- סדר החטיבות - DIVISIONS
- כתיבה באזורים B,A
- סימני פיסוק והפרדה

ב. בשורה 5 מוגדר שדה אלמנטרי בודד בקוד 77, אשר מכיל ערך התחלתי המיועד לשמש ככותרת לשורת הנתונים, שתודפס במהלך התכנית.

ג. בשורה 6 מופיעה הגדרה של נתון קבוצתי, הכוללת את <u>כל</u> השדות, המיועדים להדפסה. הנתונים האלמנטריים שלו הם KARTIS ו-DATE. אל שני שדות אלה נקרא את הנתונים כמפורט בהמשך.

ד. בשורה 8 מוגדר שדה של 3 תוי רווח להפרדה בין שני שדות הנתונים, בשורת ההדפסה.

ה. בשורה 10 מוגדרת חטיבת התכנית המכילה את כל הוראות הביצוע, ובה סעיף אחד, ששמו ניתן בשורה 11.

ו. מהלך התכנית כולל: קריאת כרטים מנוקב לשדה KARTIS (שורה 12), הדפסת השדה MESSAGE במסוף של המפעיל (שורה 13), קריאה של שדה תאריך, שיוזן ע"י המפעיל

(שורה 14), ולבסוף הדפסת השדה KARTIS-DATE כולו (שורה 15), המכיל את השדות KARTIS ו-DATE.

ז. בשורה 16 ישנה ההוראה האחרונה בתכנית, והיא גם ההוראה האחרונה שתתבצע. משמעותה - הפסקה בביצוע התכנית והעברת הפיקוח על מערכת ההפעלה.

7.6 שפת התכנות PL/I

שפת התכנות PL/I (PROGRAMMING LANGUAGE I) פותחה באמצע שנות השישים, על ידי קבוצת אנשים, שחלקה היה שייך לחברת יבמ וחלקה השתייך לאגודה של משתמשי ציוד יבמ בארה"ב. מטרת הקבוצה היתה ליצור שפת תכנות, אשר תכיל מצד אחד את כל התכונות "הטובות" של שפת הקובול (השפה "המסחרית") ושל שפת הפורטרן (השפה "המדעית"), ומצד אחר - גם אפשרויות תכנות נוספות, שאינן קיימות בשתי השפות האלו.

תכונות השפה:

1. הסימנים המותרים לשימוש: כל הסימנים, אשר מותרים בקובול ובנוסף לכך, NO (לא), OR (או), AND (וגם), %, #.

2. ההוראה (STATEMENT). כדוגמת ההוראות בשפות האחרות מתחלקות ההוראות לסוגים אחדים, לדוגמה:

הוראת השמה	A=B+C;
הוראת קלט	GET DATA (A,B);
הוראת פלט	PUT DATA (X,Y);
הוראת הסתעפות	GO TO END; ועוד.

אלו הן מספר דוגמאות, אשר אינן מייצגות את כל האפשרויות לכתיבת הוראות. יש לשים לב, שב-PL/I ההוראה מסתיימת בסימן ";" ולא בנקודה כמו בקובול.

3. פונקציות בנויות

פונקציות בנויות (BUILT-IN FUNCTIONS) הן חלק בלתי נפרד של שפת ה-PL/I. פונקציה היא קטע של תכנית, המקבלת ערך או קבוצת ערכים, ומפיקה מהם ערך חדש. למשל, פונקצית השורש הריבועי (SQRT) מקבלת ערך מסוים, ומחזירה את השורש הריבועי שלה. הפונקציות הבנויות ב-PL/I מיועדות להקל על

התכניתן, בכך שהן משמשות לו כלי מוכן, בדוק ונוח בתכנות. דוגמאות לפונקציות בנויות ב-PL/I:

SQRT - לחישוב שורש ריבועי

SIN - לחישוב הסינוס

LOG - למציאת הלוגריתם

SUM - למציאת סכום של איברי מערך, ועוד

.

השימוש בפונקציות פשוט מאד: די לקרוא לפונקציה בשם ולציין את הערך הנמסר לה לעיבוד. לדוגמה:

```
RESULT = SQRT(NUM)
```

כתוצאה יחושב השורש הריבועי של המשתנה NUM ויושם בשדה בשם RESULT. השימוש בפונקציות קיים גם בפורטרן, אך האפשרויות ב-PL/I רבות ומגוונות יותר, ולא נעמוד על כך כאן.

4. <u>ברירות מחדל</u>

שפת PL/I מצטיינת באחת התכונות החביבות במיוחד על תכניתנים - הכתיבה התמציתית. מותר לקצר בכתיבת מלים מסוימות (PROC במקום PROCEDURE, למשל) וגם אפשר לא לכתוב דברים מסוימים. ההנחה היא, שהמהדיר של PL/I יוסיף את הערך הדרוש, כאשר הוא משתמש בטכניקה של <u>ברירת מחדל</u> (DEFAULT OPTION). ברירות מחדל מייצגות את האלטרנטיבה המקובלת והשכיחה ביותר מבין כמה אלטרנטיבות, ואם לא מוגדר אחרת, המהדיר "יניח מראש" שהתכניתן מתכוון לאלטרנטיבה השכיחה.

דוגמה לברירות מחדל: בהגדרת נתון מספרי - אם לא נאמר אחרת, המהדיר יניח שמדובר בנתון <u>דצימלי</u>.

5. <u>מבנה התכנית</u>

ההוראות ב-PL/I נכתבות בטופס קידוד, המחולק לשורות ובכל שורה 80 עמודות. הכתיבה חופשית, מותר לכתוב בשורה בכל מקום בין הטורים 2-72. טורים 73-80 משמשים לזיהוי התכנית. טור 1 אינו בשימוש.

התכנית עצמה מורכבת <u>מגוש</u> (BLOCK) אחד או יותר. כל גוש הינו אוסף של הוראות, אשר מתחיל בהוראה PROCEDUCE ומסתיים בהוראה END.

דוגמה לתכנית קצרה ב-PL/I:

```
שורה
----
1  TEST : PROCEDURE OPTIONS (MAIN);
2         DECLARE (ITEM) CHARACTER(10);
3         DECLARE (UNITS, PRICE) DECIMAL;
4  START: GET LIST (UNITS, PRICE);
5         IF UNITS > 100 THEN STOP;
6         ELSE VALUE = UNITS * PRICE;
7            PUT SKIP LIST (ITEM, UNITS, PRICE, VALUE);
8            GOTO START;
9         STOP;
10 END TEST;
```

בשורה 1 נאמר שהתכנית מורכבת מגוש בשם TEST, וזוהי התכנית המרכזית (MAIN).

בשורה 2 מגדירים את השדה "פריט" (ITEM).

בשורה 3 מגדירים את השדות PRICE, UNITS.

שורה 4 - הוראת GET קוראת מדי פעם שני ערכים, יחידות ומחירים.

שורה 5 - אם הערך של UNITS עולה על 100, יש לעצור את ביצוע התכנית,

שורה 6 - ולא - יש לחשב ערך, שהוא מכפלת המחיר בכמות היחידות.

שורה 7 - מדפיסים את שם הפריט, מחיר, כמות וערך.

שורה 8 - חוזרים לנקודה START, לקריאת רשומה חדשה.

שורה 9 - עוצרים את הביצוע.

שורה 10 - סוף התכנית.

7.7 מחולל דו"חות - RPG

שפת RPG (REPORT PROGRAM GENERATOR) נוצרה בשנות השישים כדי להפיק את פלט המחשב בצורה, שתהיה נוחה לאנשים בעבודתם היומיומית. צורת פלט זו הינה הדו"ח ושם השפה "RPG" משמעותו <u>מחולל דו"חות.</u>

מי זקוק לדו"חות? - כל אחד. מנהלי חשבונות, מחסנאים, תקציבנים, מנהלים, מפעלים, בעלי בתי-עסק ואנשי מינהלה. כל

אלה ועוד רבים אחרים מעוניינים לעתים תכופות בדו"חות מודפסים. היישומים הקשורים בדו"חות אינם מסובכים מבחינת פעילות החישוב, אך גורם חשוב הוא הפלט המודפס. הדו"ח המודפס צריך להיות בעל צורה נאה ומסודרת, עם כותרות ברורות ומפורטות, סיכומי ביניים וסיכומים סופיים, עמודות, מרווחים בין קבוצות דיווח שונות ועוד.

הדרישה לחישובים פשוטים יחסית מצד אחד וזמינות זיכרון קטן מצד אחר עשו את ה-RPG למתאימה במיוחד <u>למחשבים קטנים</u> (MINI-COMPUTERS). אפשרות השימוש במחשב קטן, אשר נמצא לרוב במפעלים קטנים או בינוניים, שאצלם ההוצאה הכספית על עיבוד נתונים חייבת להיות נמוכה עד כמה שאפשר, הביא לשימוש נרחב במחולל הדו"חות בתחום זה של השוק.

תכונה נוספת של RPG - פשטות התכנות. שפת RPG שונה משפות התכנות האחרות, בכך שבמקום לכתוב הוראות שונות על טופס קידוד, כותב התכניתן שורה של מלות תיאור <u>והגדרות</u> (SPECIFICATIONS) על טפסים מיוחדים.

קיימים 4 סוגי טפסים עיקריים, ו-2 משניים. להלן תיאורם:

(1) טופס F   FILE DESCRIPTION - משמש לתיאור של קובצי הקלט והפלט ואמצעי האחסון הקשורים בהם.

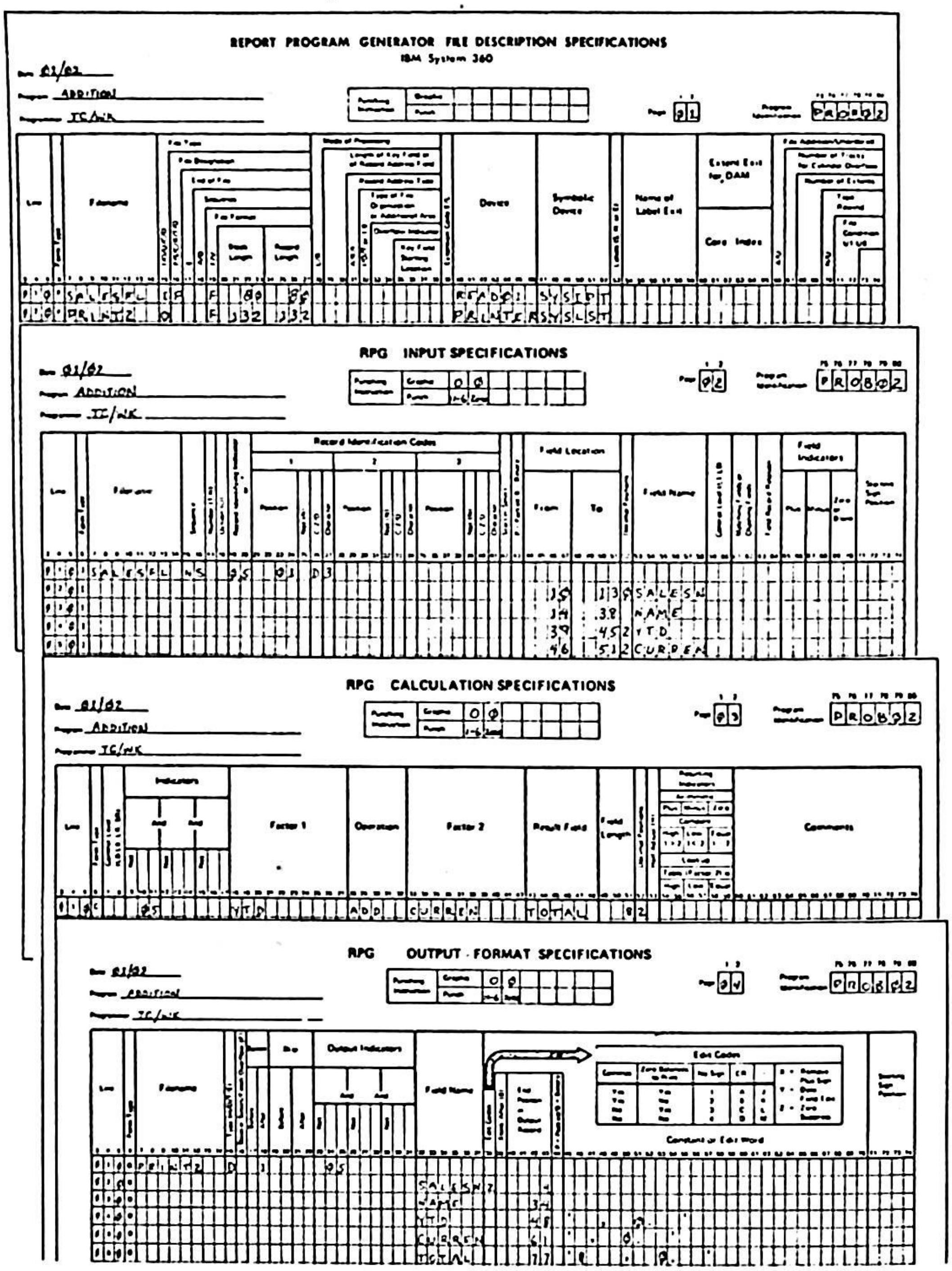
(2) טופס I   INPUT - בו מתארים את רשומות הקלט והפלט עם השדות השונים ומפרטים את התכונות שלהם (שדה מספרי, אלפביתי, אורך וכו').

(3) טופס C   CALCULATION - התכניתן מציין את הפעולות, שתתבצענה על הנתונים (פעולות חשבוניות, הזזת שדות וכו') ובדיקת התוצאות, בתוך שימוש בשדות עזר.

(4) טופס O   OUTPUT - בטופס זה יינתן תיאור מדויק של הדו"ח המופק עם פירוט הכותרות, מיקום מדויק של כל שדה והוראות עריכה לגבי שדות מספריים (למשל מיקום הנקודה העשרונית, ביטול אפסים משמאל למספר וכד').

(5) טופס E   FILE EXTENSION - כאן ניתן לציין נתונים נוספים על קבצים וטבלאות.

(6) טופס L   LINE COUNTER - בעזרת הגדרות על טופס זה ניתן לאחסן את הדו"ח הנ"ל באמצעי אחסון מגנטיים ולהדפיסו מאוחר יותר.



ציור 4-7 - תכנית RPG

7.8 שפות הידברותיות

לשפות התכנות, שתוארו עד כה בספר זה מכנה משותף: הן שפות תכנות לעיבוד באצווה (BATCH). כלומר, התכניתן כותב את התכנית בטופס, אשר מועבר בהקלדה לאמצעי הזנה למחשב. התכנית חייבת לעבור את שלב ההידור (COMPILATION), שבו מוסבות ההוראות של השפה להוראות מכונה. התוצאה נשמרת בספריית התכניות כתכנית מוכנה לעיבוד. התכניתן מריץ את התכנית לשם ניפוי שגיאות, מתקן הוראות כלשהן, מקליד לשם הזנה למחשב ועובר שוב את שלבי ההידור, ניפוי השגיאות וריצת הניסוי עד למצב שבו התכנית מוכנה לעבודת ייצור.

קיימת קבוצה נוספת של שפות, השונות מן הקבוצה הקודמת בשני היבטים עיקריים:

- גמישות התכנות בשפה.
- תרגום התכנית לשפת מכונה.

גמישות השפה מתבטאת בכך, שהתכניתן יכול לכתוב את קטעי התכנית במפוזר, והם נאספים כנדרש ע"י המהדיר או המפענח. אין כאן "קשיחות" של תכנות קובול, למשל.

הזנת התכנית נעשית בצורה **הידברותית** (CONVERSATIONAL, INTERACTIVE), כעין שיחה בין האדם-התכניתן לבין המחשב, אשר מגיב בו במקום על כל פעולה. מכאן מובן שהתכנות נעשה באמצעות מסוף, אשר מאפשר את **הקשר אדם-מכונה** (MAN-MACHINE INTERACTION).

תרגום התכנית לשפת מכונה נעשה בעזרת **תכנית מפרשת** (INTERPRETER). תכנית תרגום זו מתרגמת כל שורה ברגע שהיא מוזנת למחשב. יתרה מזו, התרגום כולל גם בדיקת שגיאות, ועל כן ניפוי של שגיאות תחביר יכול להתבצע בו במקום.

בעשור האחרון נפוצו שתי שפות הידברותיות כאלה: BASIC ו-APL. בעוד שלשפת BASIC יוקדש פרק שלם בספר הזה, הרי ששפת APL מוצגת כאן בקצרה. גם לשפות COBOL, FORTRAN ו-PL/I, אשר הוזכרו קודם פותחו ניבים, אשר מאפשרים תכנות הידברותי. בכך הן מצטרפות לקבוצת השפות הזו.

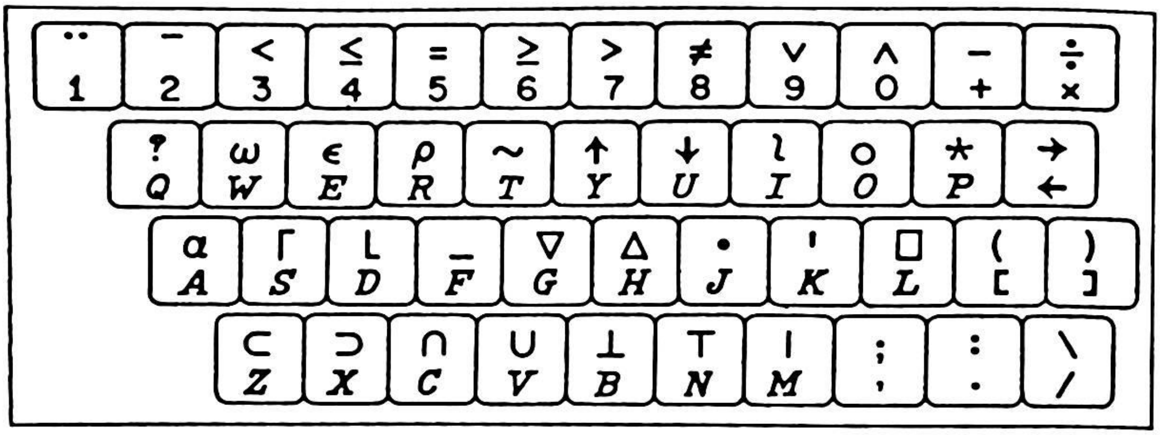
7.9 APL

APL (A PROGRAMMING LANGUAGE) נוצרה בתחילת שנות ה-60 על ידי KENNETH IVERSON כעבודת דוקטורט שלו. כוונת המחבר היתה ליצור שפת תכנות נוחה לפתרון בעיות באופן מיידי. במשך השנים נמצא, ש-APL מתאימה מאוד לפתרון בעיות מתמטיות, והשימוש העיקרי בה כיום הינו בתחום זה ובתחום פתרון אלגוריתמים.

תכונות שפת APL :

1. שפת APL בנויה להפעלה בסביבה של **שיתוף זמן** (TIME SHARING, ראה פרק 11) באמצעות מסופים בעלי אביזרים ומקלדת מיוחדים.

2. בשפת APL מותר להשתמש באותיות ה-ABC, בספרות 0-9 ובמערכת סימנים מיוחדת כפי שמוצג בציור 7-5.



ציור 5-7 - מקלדת עבור APL

3. ההוראות בשפת APL מוזנות למחשב דרך המסוף ומפורשות (INTERPRETED) ע"י תכנית מיוחדת (APL INTERPRETER), בזו אחר זו.

4. תכנות בשפת APL מתבצע בשטח עבודה מיוחד בזיכרון המחשב, הנקרא WS (WORKSPACE). שטח עבודה זה מכיל תכניות APL, נתונים ואמצעי בקרה שונים. שטח העבודה אינו מוגבל לתכנית אחת בלבד, אך לעתים נוח לעבוד על נושאים שונים בשטחי עבודה שונים. את שטחי העבודה אפשר לשמור לשימוש חוזר, כי הרי הם מכילים תכניות ונתונים.

כאן יש להדגיש את ההבדל בין APL ושאר השפות העיליות, שהוצגו קודם לכן. בעוד שתכניות, אשר נכתבו בשפות התכנות האחרות, תורגמו לתכניות בשפת מכונה בתוך כדי תהליך הידור, והתבצעו במחשב לאחר מכן, הרי תכנית בשפת APL מתבצעת בדרך שונה. כל הוראה, המוזנת למחשב במסוף, הופכת להוראה בשפת מכונה ומתבצעת בו-ברגע, לפני שההוראה הבאה בתכנית מוזנת למחשב. ישנה גם אפשרות להזין את כל ההוראות יחד, כתכנית, ורק אח"כ להפעילן.

5. פונקציות בנויות (BUILT IN FUNCTIONS) הן חלק בלתי נפרד של שפת APL וע"י השימוש בהן מפשטים ומייעלים את כתיבת התכנית.

להלן תכנית קצרה ב-APL.

```
    PRICE <--- 13.25  6.75  .95  1.25                  1
    QUANTITY <--- 53  3  141  503                      2
    COST <--- PRICE X QUANTITY                         3

    MAXCOST <--- ⌈/ COST                               4
    MAXCOST                                            5
702.25                                                 6
    MINCOST <--- ⌊/ COST                               7
    MINCOST                                            8
 20.25                                                 9
    COST                                              10
702.25  20.25  133.95  628.75                         11
```

ציור 7-6 - "שיחת" APL טיפוסית

הסבר לשיחה ב-APL לפי ציור 7-6

שורה 1 בשורה זו ניתנת הוראה למחשב להגדיר קבוצת שדות בזיכרון, בשם PRICE, ולאחסן בהם את המחירים, המופיעים מימין למלה PRICE. PRICE הינו וקטור בן 4 איברים.

שורה 2 הוראה דומה לקודמת, אך שם הקבוצה QUANTITY, והקבוצה מקבלת את הערכים הרשומים מימין למלה הנ"ל.

שורה 3 זו הוראת השמה, המוכרת משפות פורטרן או PL/I. ההוראה מורה למחשב לכפול את איברי הקבוצה PRICE באיברי הקבוצה של QUANTITY בהתאמה ולאחסן בקבוצה בת 4 איברים בשם COST.

שורה 4 המחשב צריך לבחור את המכפלה הגבוהה ביותר מבין איברי COST ולשים בשדה MAXCOST.

שורה 5 מבקשים מן המחשב להציג את תוכן שדה MAXCOST.

שורה 6 הערך 702.25 מוצג כתוכן של MAXCOST.

שורות 7-9 דומות לשורות 4-6, אך כאן היה צריך למצוא ולהציג את הערך הנמוך ביותר בקבוצת הערכים בשם COST.

שורה 10 מבקשים להציג את ערכי COST.

שורה 11 הצגת האיברים של COST.

פרק 7 - שאלות

1. הסבר את המונחים: שפה עילית, שפת סף, שפה יעודת מכונה.
הדגם כל אחת מהן.

2. מהי הוראת השמה? הדגם הוראה כזו בפורטרן וב-PL/I.

3. מהי הוראת בקרה?

4. מהן מלים שמורות? באיזו שפת תכנות הן מוגדרות?

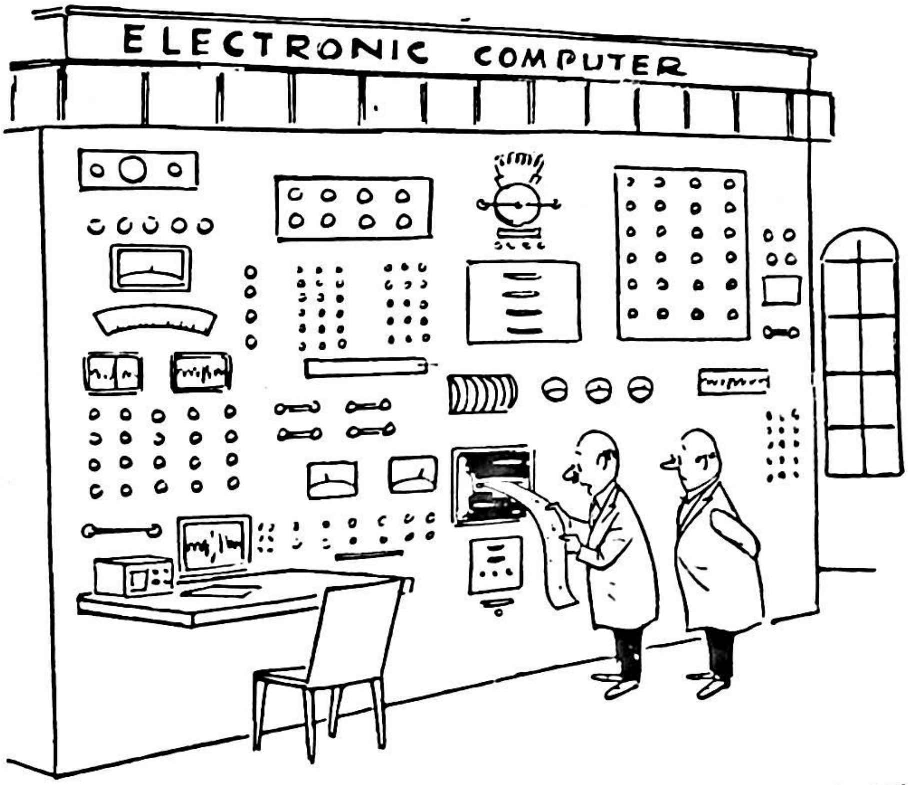
5. נסה לכתוב תכנית בפורטרן לתרגיל גאי, המופיע בסוף פרק 6.

6. מה מאפיין שפות הידברותיות?

7. מהן פונקציות בנויות? באילו שפות הן מופיעות?



לכל בעיה — פתרון!



"אפסיק תפוקה, רוצה חופשה!"

פרק 8 – מארכות הפעלה

בפרק הקודם הוסבר כיצד נכתבת תכנית המקור, ואילו שלבים היא עוברת, עד שהיא הופכת לתכנית, המוכנה לביצוע. נפנה עתה אל מערכת המחשב, בה מתבצעת התכנית.

מערכות המחשב הראשונות, מלפני 30 שנה, היו קטנות ופשוטות. יחידת הזיכרון היתה בגודל 4K או 8K, לכל היותר. במחשבים ההם אפשר היה לבצע רק תכנית בודדת, ובעת הביצוע נועד תפקיד חשוב מאוד <u>למפעיל המחשב</u>. הוא קיבל מידי התכניתן את התכניות לביצוע, בצורה של חבילת כרטיסים, או על גבי סרט מגנטי. אל התכניות צורפו הוראות שונות, שהתייחסו לסוג המהדיר המבוקש, לכונני הסרטים הדרושים, לסוג הנייר עליו יודפס הפלט ועוד.

לאחר שהמפעיל קרא את ההוראות, הוא הזין את המהדיר לזיכרון, הרכיב את הסרט הנכון על הכונן, והכניס נייר מתאים במדפסת. לבסוף הזין את התכנית לביצוע, ולחץ על כפתור START.

לאחר ביצוע התכנית, היה המפעיל אוסף את הפלט המודפס, ויחד עם הסרטים והכרטיסים שם אותו בתא של העבודות הגמורות. אחרי כן החל בהכנות לקראת הביצוע של התכנית הבאה בתור. תמונה פסטורלית זו הופרעה לעתים על ידי צלצול הטלפון או בואה של עגלת התה. אז היה מפסיק המפעיל את עבודת המחשב, שותה תה, אוכל כריך וחוזר לעבודתו. מצב זה, כמובן, לא השביע את רצון האחראים על יחידת המחשב. ביחידות המחשב נאלצו לעבוד שעות נוספות, ולרב גם עבדו במשמרות לילה, ולמרות כל זה לא הספיקו לבצע את כל העבודות בזמן הדרוש.

בשנות השישים נעשו נסיונות משמעותיים להתגבר על הבעיות השונות שהוזכרו. המאמצים הופנו לכיוונים הבאים:

- מציאת שיטה אחידה למתן הוראות למפעיל;
- צמצום, עד למינימום, של מעורבות המפעיל בעבודת המחשב;
- מציאת שיטה למעבר מהיר בין עבודה אחת לאחרת, ללא התערבות המפעיל, עד כמה שאפשר.

להלן נסקור את הפתרונות, שנמצאו במשך הזמן לבעיות אלו, וכן את החידושים הנוספים, שמטרתם להביא ליעילות מירבית בעבודתה של מערכת המחשב כולה.

8.1 תכנית פיקוח על עבודות

כזכור, אל כל תכנית מחשב צורפו ההוראות המיוחדות של התכניתן, אשר הודיעו למפעיל ולמחשב את שם התכנית, שם התכניתן, שם המהדיר שהתכנית זקוקה לו, ומהם אמצעי אמצעי הקלט והפלט הדרושים לביצוע. עתה נשים לב לשתי עובדות חשובות:

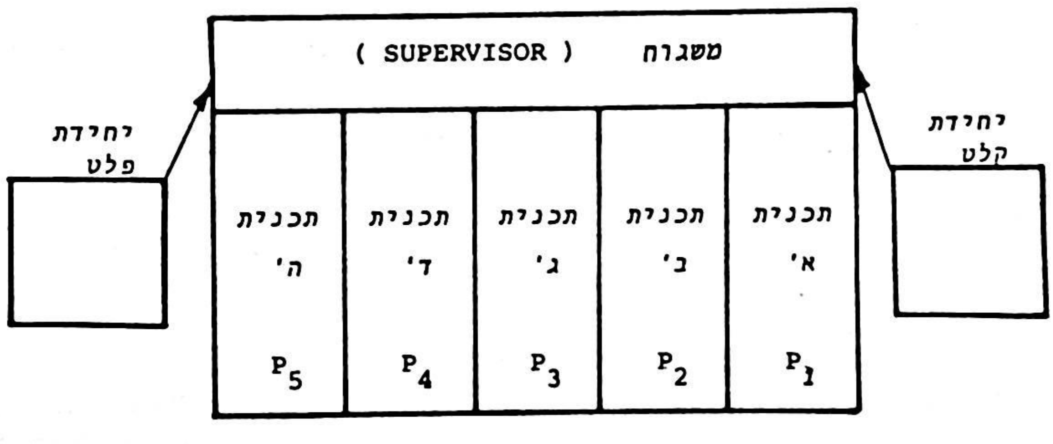
1. הוראות ההפעלה של כל תכנית אמורות להיקרא על ידי המחשב, לפני שהמהדיר נקרא לפעולה;

2. לפני כל תכנית ניתנו הוראות כמעט זהות (סוג הנייר, סוג אמצעי קלט/פלט וכוי), ולכן פותחה **שפת פיקוח לעבודות מחשב** (JOB CONTROL LANGUAGE). שפה זו מאפשרת מתן הוראות מקדימות לביצוע, הן בדבר סוג האמצעים הדרושים והן בדבר הקבצים, העומדים להשתתף בעיבוד (שם הקובץ, מבנה הרשומות, האם יש לשמור על הקובץ לשימוש מאוחר יותר וכוי).

למעשה, בדומה למהדירים, נכתבה תכנית מיוחדת על ידי היצרנים, שתפקידה להבין את שפת הפיקוח ולבצע את ההוראות, שנמסרו בה. השם המקובל של תכניות מסוג זה הוא **תכנית פיקוח לעבודות** (JOB CONTROL PROGRAM). לתכנית זו גם תפקיד נוסף: פיקוח על העבודות הנכנסות לביצוע. היא דומה, למעשה, לשוער העומד בפתח המחשב ומודיע על האורח, שנכנס לביצוע, מקבל אותו, ודואג לו לאחר שסיים את תפקידו שם. תכנית זו הלכה והתרחבה במשך השנים, והפכה להיות חלק ממערכת ההפעלה, עליה ידובר להלן.

8.2 תכנות מרובב

בדור השני והשלישי של המחשבים הלך וגדל מספר הבתים בזיכרון. מזיכרון צנוע של 4K התקדמו יצרני המחשבים אל יחידות זיכרון בעלות 16K, 32K, ועד 256K. באמצעות זיכרון בגודל כזה אפשר להגשים רעיון ישן: הכנסת מספר תכניות לביצוע בתוך יחידת זיכרון אחת.



ציור 8-1 - חלוקת הזיכרון לקטעים

כבר במחשבים של שנות השישים אפשר היה להקצות בזיכרון המרכזי קטעים או **מחיצות** (PARTITIONS), ובכל אחד מהם לאחסן תכנית שונה (ראה ציור 8-1). בקטע P1 תאוחסן תכנית א', ב-P2 תכנית ב', וכך הלאה.

עתה נשאלת השאלה, האם התכניות יכולות לעבוד **ביחד**, בעת ובעונה אחת? התשובה לשאלה זו אינה כה פשוטה. הבה ונבחן אותה:

בכל תכנית ישנם שני סוגים של הוראות: הוראות חישוב, שמתבצעות על ידי היע"מ, והוראות קלט/פלט, אשר מתבצעות על ידי יחידות ק/פ. **היע"מ יכול לבצע רק פעולת חישוב אחת בזמן מסוים** ולכן לא ניתן לבצע, למשל, שתי הוראות חישוב בעת ובעונה אחת. אך ייתכן מצב, אשר קיים בד"כ, שבו תכנית א' מבקשת לבצע הוראת ק/פ, ואילו תכנית ב' מבקשת לחשב דבר מה. במצב כזה, **שתי התכניות** תקבלנה שירותים מהמחשב באותו זמן: יחידות ק/פ תשרתנה את תכנית א', והיע"מ - את תכנית ב'. יתרה מזו, ידוע, שפעולת ק/פ נמשכת זמן רב יותר (מספר אלפיות שניה), מאשר פעולת חישוב (שזמן הביצוע שלה נמדד במליוניות השנייה). לכן יחידת החישוב מספיקה לשרת גם את תכנית ב' ואפילו את תכניות ג' ו-ד', בזמן שמתבצעת פעולת ק/פ בעבור תכנית א'.

באופן כללי ניתן לומר, ששילוב פעולות קלט/פלט עם ביצוע הוראות חישוביות יכול להעלות את תפוקת המערכת כולה. שיטת עבודה שבה מופעלת בזיכרון המחשב יותר מתכנית אחת בעת ובעונה אחת, נקראת **תכנות מרובב** (MULTI-PROGRAMMING). השימוש בשיטה זו תורם באופן משמעותי לניצול יעיל יותר של משאבי המערכת.

8.3 תכנית המשגוח

בשיטת העיבוד המרובב ניתן לאחסן ולהפעיל בזיכרון הראשי מספר תכניות. במחיצה P1 תימצא תכנית משכורת (נקרא לה בשם תכנית א'), במחיצה P2 תהיה תכנית לניהול מלאי וכו'. במצב זה יכולות להתעורר מספר בעיות, שעל כמה מהן נרמוז כאן בקצרה:

1. האם ניתן להפריד הפרדה מוחלטת בין המחיצות כדי שתכנית אחת לא תפריע לחברתה?

2. נניח שתכנית א' מדפיסה תלושי משכורת ותכנית ב' מדפיסה רשימת פריטים במלאי. מה יקרה, כאשר נרצה להדפיס את שני הדו"חות בעת ובעונה אחת במדפסת היחידה של המערכת?

3. הביצוע של תכנית ג' הסתיים. האם יודע המפעיל על כך, ודואג להכנסת תכנית חדשה לביצוע? ואם כן, האם הוא יודע לאיזו מחיצה? כיצד נמנע המתנה או עיכוב בתפעול המערכת?

4. בדרך כלל התכניות אינן מאוזנות מבחינת הרכב ההוראות שבהן. ישנן תכניות בהן, באופן יחסי, מתבצעות יותר הוראות ק/פ מאשר הוראות חישוביות, והן נקראות בשם תכניות I/O-BOUND. לעומתן, קיימות תכניות, בהן מתבצעות יותר הוראות חישוב מהוראות ק/פ, והן נקראות תכניות CPU-BOUND.

כדי להתגבר על הבעיות הללו, היתה דרושה תכנית מיוחדת, שתפקידה העיקרי - פיקוח על ביצוע תקין ושוטף של התכניות השונות. תכנית פיקוח כזאת נקראת <u>משגוח</u> (SUPERVISOR). חלק מתכנית זו, המסומנת באותיות SPVR בציור 8-1, שוכנת כל העת בזיכרון, ושולטת על הנעשה במחיצות השונות. היא דואגת, שתכנית אי במחיצה מסוימת לא תפריע לשכנותיה; שביצוע תכנית יופסק אם היא נכנסה ל-LOOP, ונצלה יותר מדי זמן של המערכת. המשגוח הינו רק חלק מקבוצה שלמה של תכניות, ששמן הכולל הוא <u>מערכת הפעלה</u> (OPERATING SYSTEM). תכניות אלו פותחו על ידי יצרני המחשבים, והן מסופקות יחד עם המחשב, בעת קנייה או שכירות.

קיימות מספר הגדרות למערכת ההפעלה, ואנו בחרנו באחת מהן:

"מערכת הפעלה היא קבוצת תכניות, שתפקידן לנהל את כל משאבי מערכת המחשב בצורה אופטימלית, ולספק שירותים שונים למשתמש". הבה נראה אילו תכניות נכללות במערכת הפעלה, ומהם השירותים, שהן מספקות.

<u>8.4 המרכיבים של מערכת ההפעלה</u>

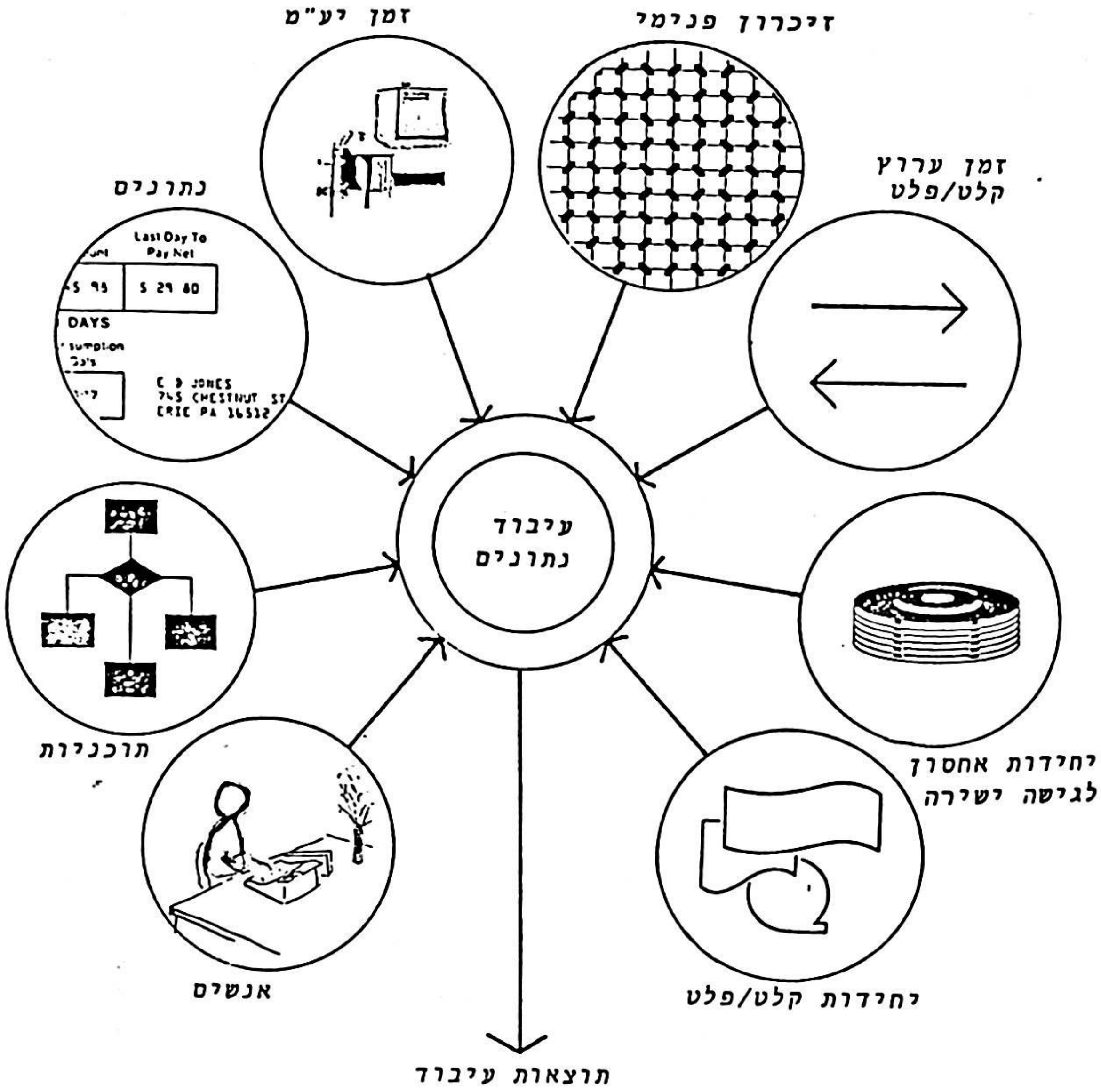
מן הראוי לציין, שקיימים סוגים שונים של מערכות הפעלה. כל יצרן מחשבים מספק בדרך כלל עם המחשב, שהוא מוכר, גם מערכת הפעלה, אשר מתאימה לאותו מחשב, או לאותה משפחת מחשבים. אולם למרות המספר הגדול של סוגי מחשבים ושל סוגי מערכות הפעלה, ישנו דמיון בין מערכות ההפעלה, כי ביסודו של דבר המטרות העיקריות - קרי פיקוח על ביצוע, ניהול משאבים ומתן שירותים - קיימות כמעט בכולן.

תכניות של מערכת ההפעלה מתחלקות בדרך כלל לשני סוגים: תכניות פיקוח ותכניות ביצוע.

<u>8.4.1 תכניות פיקוח</u>

בקבוצה הראשונה, הנקראת <u>תכניות פיקוח</u> (CONTROL PROGRAMS), נכללות בדרך כלל שלוש תכניות, אותן נתאר כאן.

1. <u>תכנית טעינה ראשונית</u> (IPL) או בשמה המלא: INITIAL PROGRAM LOADER. היא תכנית קצרה מאד, אשר <u>נמצאת</u>



ציור 2-8 - המשאבים הדרושים לתהליך עיבוד הנתונים

על דיסק מגנטי או תקליטון. בתחילה, לפני השימוש בדיסקים שמרו אותה בכרטיסים או בסרט מגנטי, והיא נטענה ממנו בצורה מיוחדת. לעתים רחוקות נמצא את תכנית IPL צרובה <u>בזיכרון בלתי מחיק</u> (ROM). תכנית IPL נטענת לזיכרון ע"י המפעיל, וזאת למעשה <u>הפעולה הראשונה</u>, שנעשית לאחר שמפעילים את מתג החשמל של המחשב, ומכאן שם התכנית.

תפקיד ה-IPL מצומצם, אך חשוב. הוא מתבטא במספר קטן של פעילויות: בדיקת המחשב, "ניקוי" הזיכרון, הפעלת שעון פנימי של המחשב וטעינה של תכנית המשגוח לזיכרון. ביצוע ה-IPL יכול להמשך עשרות שניות עד למספר דקות, בהתאם למערכת המחשב.

2. <u>תכנית המשגוח</u> (SUPERVISOR) הינה אוסף של תכניות קטנות יותר (הנקראות גם שגרות), שלכל אחת מהן תפקיד מיוחד ומוגדר. השגרות העיקריות של המשגוח מטפלות במצבים הבאים:

- <u>פיקוח על ניצול יעיל של הזיכרון</u>, בתוך כדי שמירה על עקרון ההפרדה בין המחיצות. הודות לכך אפשר לעבוד בשיטת התכנות המרובב, ועל ידי כך להגביר את תפוקת העבודה של המערכת.

- <u>טעינת תכניות שונות לביצוע.</u> תכנית המשגוח, אשר שולטת על אחד מהמשאבים החשובים של המערכת, כלומר על הזיכרון הראשי, מבחינה איזו מחיצה מתפנה. אז היא טוענת למחיצה שהתפנתה תכנית, שביצועה נדרש באותו זמן.

- <u>רשות להשתמש ביע״מ.</u> מכיון שישנן תכניות רבות בזיכרון, הן מתחרות ביניהן על השימוש במשאב עיקרי נוסף של מערכת המחשב, היע״מ, אשר בעזרתו מתבצעים החישובים והעיבודים השונים. המשגוח מופיע כאן כעין שוטר תנועה, המתיר את השימוש ביע״מ לתכניות השונות. כל תכנית, הרוצה לחשב דבר מה, תעשה זאת בתיווכו של המשגוח בלבד.

- <u>פיקוח על פעולות קלט/פלט</u> בעזרת טבלאות מיוחדות. הוראות הפיקוח, אשר צורפו אל התכניות, שנקראו לביצוע, התייחסו, בין היתר, לשימוש באחד מהמשאבים החשובים של המערכת, יחידות ק/פ. תכנית מתאימה של המשגוח מנהלת רישום מדויק, איזו תכנית מאיזו מחיצה משתמשת ביחידת ק/פ מסוימת. אגב כך היא שומרת, שלתכנית אחרת לא תהיה גישה אל אותה יחידה, עד לסיום השימוש בה.

- <u>טיפול בתקלות חמרה ותכנה.</u> בתוך כדי ביצוע התכנית יכולות לקרות תקלות שונות. נמחיש זאת בשתי דוגמאות:

  – <u>תקלה בחמרה</u>
  תוך כדי הדפסת דו״ח אזל הנייר במדפסת. אמצעי בקרה, הממוקם במדפסת (עין אלקטרונית, למשל) מודיע על כך לתכנית המשגוח. <u>שגרה</u> מתאימה שולחת הודעה למצג של המפעיל, ובינתיים "מקפיאה" את ביצוע התכנית, אשר היתה באמצע תהליך ההדפסה (התכניות ביתר המחיצות ממשיכות לעבוד, כמובן). המפעיל מזין במדפסת נייר מתאים ומודיע למשגוח, שמצדו הכל מוכן להדפסה. שגרת המשגוח מקבלת את ההודעה, ו"משחררת" את התכנית שהוקפאה.

  – <u>תקלת תכנות.</u> בעת ביצוע של תכנית מסוימת מגיע המחשב אל פעולות חילוק בין שני מספרים, א׳ וב׳. אולם ערכו של ב׳ שווה לאפס. חילוק א׳ וב׳ ייתן מספר אינסופי, וזה נוגד את כללי הביצוע במחשב. שגרה מתאימה של המשגוח מפסיקה את ביצוע התכנית

הזאת לאחר הודעה מתאימה, שנשלחת אל מצג המפעיל, והמערכת עוברת לביצוע תכנית חדשה. אפשר גם להורות על עקיפת התקלה, לאחר הודעה מתאימה וציון השגיאה, ולהמשיך בביצוע.

• <u>קבלת הודעות ממצג המפעיל ומשלוח הודעות אליו.</u> מצג המפעיל הינו, למעשה, הצינור העיקרי (וכמעט היחיד), שדרכו יכול המחשב להתקשר אל העולם החיצוני. דרך המצג מודיע המחשב (או ליתר דיוק המשגוח) למפעיל על התקלה במדפסת, למשל. דרך המצג יכול המפעיל להודיע למשגוח, שיש צורך לאחד שתי מחיצות, כדי לבצע תכנית גדולה במיוחד.

3. <u>תכנית פיקוח לעבודות</u> (JOB CONTROL PROGRAM) הינה התכנית השלישית בקבוצה זו. היא תוארה בפרק זה, בסעיף 8.1. נחזור בקצרה על עיקרי תפקידה:

לקרוא את <u>הוראות הפיקוח</u> (JOB CONTROL STATEMENTS), המצורפות אל התכניות ע"י התכניתן, לפענח את הדרישות לגבי השימוש ביחידות קלט/פלט ובזיכרון הראשי, ולהעבירן אל המשגוח. לאחר מכן הפיקוח עובר אל המשגוח, אשר אחראי להפעלת התכנית עד גמר הביצוע. בתום הביצוע הפיקוח עובר שוב אל תכנית פיקוח העבודות, אשר חוזרת וקוראת את הקבוצה הבאה של הוראות הפיקוח.

<u>8.4.2 תכניות ביצוע</u>

הקבוצה השנייה של תכניות מערכת ההפעלה נקראת <u>תכניות ביצוע</u> (PROCESSING PROGRAMS). הן נבדלות מן הקבוצה הראשונה בכך, שבמקום לעסוק בפיקוח, הן עוסקות בביצוע. כל תכנית בקבוצה זו מקבלת קלט מסוים, מעבדת אותו ומפיקה ממנו פלט.

נסקור כאן ארבע תכניות מקבוצה זו:

1. <u>תכניות מתרגמות, מהדירים</u> (LANGUAGE PROCESSORS, COMPILERS). תכניות אלו נסקרו בהרחבה בפרק 7 של ספר זה. נזכור שוב, שהודות למהדירים, כל תכניתן יכול לכתוב כיום תכנית בשפה עילית, הנוחה לכתיבה, ואין הוא צריך לבזבז את זמנו ולטרוח בחישובי כתובות בזיכרון ובכתיבת תכנית בשפה הבינרית. פלט המהדיר, כזכור, הוא התכנית בשפת המכונה.

2. <u>תכנית הספרן</u> (LIBRARIAN) מטפלת בספריות של המערכת, כגון ספרייה לתכניות מקור, ספרייה לתכניות בשפת מכונה, ספרייה לתכניות יישום ועוד. הספריות הן למעשה שטחים על דיסקים מגנטיים, המחוברים אל המחשב. תכנית הספרן היא, אשר יוצרת אותן ומבצעת למעשה את כל אותן הפעולות, שספרן מבצע

בספרייה: מוסיפה ומקטלגת תכניות חדשות, מבטלת תכניות ישנות, מדפיסה את רשימת התכניות ועוד.

3. <u>תכנית העורך</u> (LINKAGE EDITOR) היא אחת מתכניות השירות החשובות ביותר. תפקידה להכין לביצוע את התכנית, שעברה הידור ונמצאת במצב של שפת מכונה. הכנה זו מתבטאת באיסוף תכניות נוספות ושגרות, אשר נדרשות כדי לבנות תכנית שלימה לביצוע משימה. למשל, בעת עיבוד תכנית של ניהול מלאי אפשר לצרף מתוך ספריית התכניות את השגרה הקבועה לבדיקת הרמה של מלאי מינימום, תכנית של הדפסת דו"ח יתרות מלאי ושגרות סטנדרטיות לשם ביצוע פעולות קלט/פלט. אין צורך, שכל תכניתן יכתוב את השגרות מחדש, אלא אם הקיימות אינן מתאימות לדרישותיו. הפלט של תכנית העורך הינו התכנית הכוללת, אשר נטענת לתוך הזיכרון לביצוע.

4. <u>תכנית מיון</u> (SORT), איננה חלק ממערכת ההפעלה, אך נציין אותה בקבוצת תכניות זו, מכיון שהיא חשובה, וקיימת בכל מתקן מחשב. בכל יחידת מחשב מעבדים קבצים, ולעתים קרובות מאד מתעורר הצורך למיין קובץ מסוים לפי שדה או שדות כלשהם. תכנית המיון מאפשרת גם למזג שני קבצים, או יותר.

5. <u>תכניות המשתמש</u> (USER PROGRAMS) אינן חלק אינטגרלי של מערכת ההפעלה. הן תכניות, שנכתבו על ידי המשתמש, וצורפו אל מערכת ההפעלה במועדים שונים. אלו הן תכניות מיוחדות, שהשימוש בהן רב, ותכניות רבות נזקקות להן לעתים קרובות. הן משלימות פונקציות ניהול, בקרה, או עיבוד, אשר נדרשות במתקן מסוים, ואינן כלולות במערכות ההפעלה הסטנדרטיות.

8.5 צעדים נוספים בפיתוח

בתחילת הפרק הזה הובא תיאור של שיטות עבודה מלפני עשרים עד שלושים שנה. לאחר מכן תוארה ההתפתחות המרשימה בתכנה, אשר הביאה ליצירת מערכות הפעלה רבות, שאחת מהן תוארה בסעיף הקודם.

במשך עשרים השנים האחרונות חלה התקדמות רבה בשטח פיתוח החמרה. המחשבים נעשו מהירים יותר, וניתן לאחסן היום יותר ויותר מידע בשטח קטן יחסית ובמחיר זול לאין-ערוך מאשר לפני שתי עשרות שנים, כפי שזה מוצג בטבלה 3-8.

פיתוח מהיר זה של ציוד המחשבים הביא בעקבותיו פיתוח לא פחות מרשים בשטח התכנה, אולם יש לציין, שפיתוח החמרה לא היה הגורם הבלעדי לכך. מצד אחד הלך מספר היישומים וגדל בקצב מהיר, ומצד

מידת ההתקדמות	1984	1964	תחום / שנים
פי מליון	נמדד בננו-שניות (1:1,000,000,000 השנייה)	נמדד באלפיות שנייה (1:1000 השנייה)	זמן ביצוע הוראות
פי 4,000	64,000,000	16,000	גודל הזיכרון המרכזי (בתים)
פי 360	2,520,000,000 (IBM 3380)	7,000,000 (IBM 2311)	קיבולת דיסק ממוצע (תווים)
פי 300	3,000,000 תווים בשנייה	10,000 תווים בשנייה	מהירות העברה של נתונים אל הזיכרון המרכזי, בממוצע
פי 200,000	0.00026 סנט	50 סנט	העלות לאגירת תו אחד

8-3 - טבלה השוואתית - התפתחות החמרה בעשרים השנים האחרונות

אחר הוכפל מספר המשתמשים מידי כמה שנים. כך ניתן לראות תכניות מיוחדות ושיטות עיבוד חדישות, שמטרתן אחת: לנצל בצורה טובה יותר את הכלים, שנעשו משוכללים יותר ויותר. המאמצים הופנו לכל הכיוונים: יחידות ק/פ, יע"מ, הזיכרון המרכזי והאדם, אשר משתמש במחשב.

דוגמה טובה ליעול עבודת המחשב הוא העיבוד המרובב (MULTI-PROCESSING).

המונחים האנגליים MULTI-PROGRAMMING ו-MULTI-PROCESSING משמשים בערבובייה, אך יש להדגיש, שאין הם שמות נרדפים לאותה שיטת עיבוד, ושקיים הבדל משמעותי ביניהם.

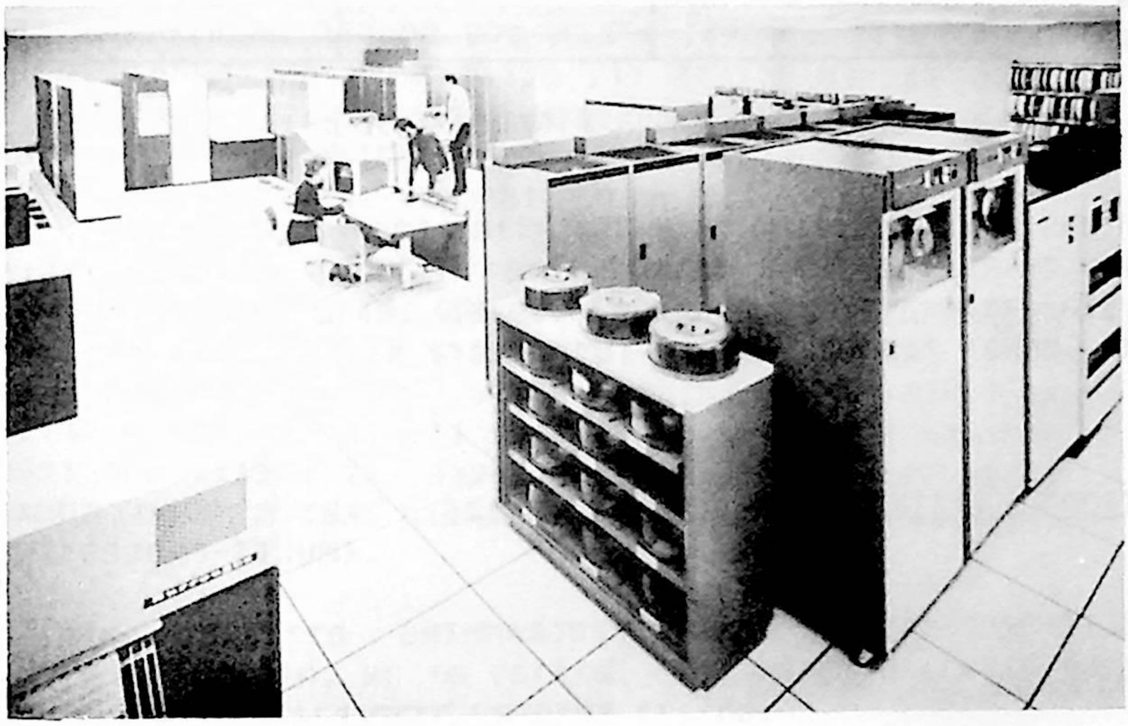
MULTI-PROGRAMMING, או תכנות מרובב, כפי שצוין בתחילת הפרק, מתייחס לשיטת עבודה, בה שתי תכניות או יותר מתבצעות במערכת מחשב אחת ותחת אותה מערכת הפעלה. ביצוע התכניות קורה ברצף, תכנית אחת בצמוד לאחרת, אך בזמנים שונים.

לעומת זאת, בשיטת MULTI-PROCESSING, או עיבוד מרובב, ההוראות של שתיים או יותר תכניות שונות מתבצעות במקביל, באותו פרק זמן על ידי שתיים או יותר יחידות עיבוד מרכזיות (CPU'S). פיקוח על הביצוע נעשה על-ידי תכנית פיקוח מרכזית אחת אשר מפקחת גם על יחידת זיכרון אחת, אשר משותפת ליע"מים השונים.

בשיטת העיבוד המרובב הבסיסית, יע"מ גדול אחד מבצע את פעולות העיבוד העיקריות, בעוד שפעולות אחרות, ניהול יחידות ק/פ, תתבצענה על ידי יע"מים קטנים יותר.

עיבוד מרובב בקנה מידה גדול מתבסס על על שימוש במספר יע"מים גדולים, דוגמת המחשב 3081 של חברת יבמ. מחשב זה מכיל שני מעבדים זהים העובדים במשותף. מודלים חדשים של מחשבים בנויים מארבעה יע"מים, כדוגמת IBM 3084, בנוסף למעבדים אחרים לשרות יחידות קלט/פלט. יתרון נוסף של מערכת זו טמון בעובדה, שאם מעבד אחד מפסיק לעבוד מסיבה כלשהי, ממשיך האחר לפעול לבדו, ומבצע בעצמו את כל הפונקציות המוטלות על המערכת.

חברת בורוז הכריזה על מערכת B6700, בה ישנם שלושה יע"מים גדולים ושלושה מעבדים נוספים לביצוע פעולות ק/פ (ראה ציור 8-4). במחשב אחר של חברת בורוז, B7700, ישנם שמונה מעבדים ושמונה מערכות זיכרון. מחשב זה תוכנן בעבור ישומי תקשורת ובסיסי נתונים.



ציור 8-4 - מחשב בורוז B6700

8.6 זיכרון בפועל

המאמצים לניצול יעיל יותר של משאבי המערכת נמשכים כל העת. במקביל לפיתוח של שיטות עיבוד יעילות, כגון העיבוד המרובב והתכנות המרובב, התמקדה תשומת הלב גם בזיכרון המרכזי. חלוקת הזיכרון למחיצות תרמה, אמנם, תרומה נכבדה לניצול יעיל של משאבי המערכת, אך עדיין נותרו מספר בעיות, שעד תחילת שנות ה-70 טרם מצאו את פתרונן. נביא שתי דוגמאות:

1. אחת המגבלות הרציניות בעבודת תכנית רגילה היתה העובדה, שהתכניתן חייב היה לקחת בחשבון את גודל הזיכרון, שעומד לרשות התכנית בזמן הביצוע. אם התכנית היתה במקרה גדולה

מן המחיצה, שמוקצית לה בזמן ריצה - לא יכלה התכנית להתבצע. במקרה כזה נאלץ התכניתן להשתמש בטכניקות שונות כדי לצמצם את התכנית, ולהתאימה לגודל הזיכרון המוקצה. מובן ששינויים כאלה גזלו זמן ממושך ומאמצים רבים.

2. בשיטת התכנות המרובב הנזכר לעיל, מחולק הזיכרון הראשי למספר מחיצות. נציין לדוגמה מערכת מחשב, שבה הזיכרון מחולק לחמש מחיצות (P1-P5), ובהן רצות חמש תכניות A,B,C,D,E. מצב זה מוצג בציור 8-5.

מערכת הפעלה				
תכנית A	תכנית B	תכנית C	תכנית D	תכנית E
40K	50K	30K	50K	20K
P1	P2	P3	P4	P5

ציור 8-5 - תיכנות מרובב - מצב א'

עתה נניח, שביצוע תכניות E,C,A הסתיים, ותכנית H, הדורשת 50K זיכרון, מחכה לביצוע. המצב החדש מתואר בציור 8-6.

מערכת הפעלה				
(פנוי)	B	(פנוי)	D	(פנוי)
P1	P2	P3	P4	P5

ציור 8-6 - תכנות מרובב - מצב ב'

מתברר, שלמרות העובדה, ששלוש מחיצות פנויות, ולמרות שסה"כ הזיכרון הפנוי מגיע לכדי 90K, תכנית H אינה יכולה להכנס לביצוע, והיא חייבת להמתין, עד שאחת המחיצות הגדולות תתפנה.

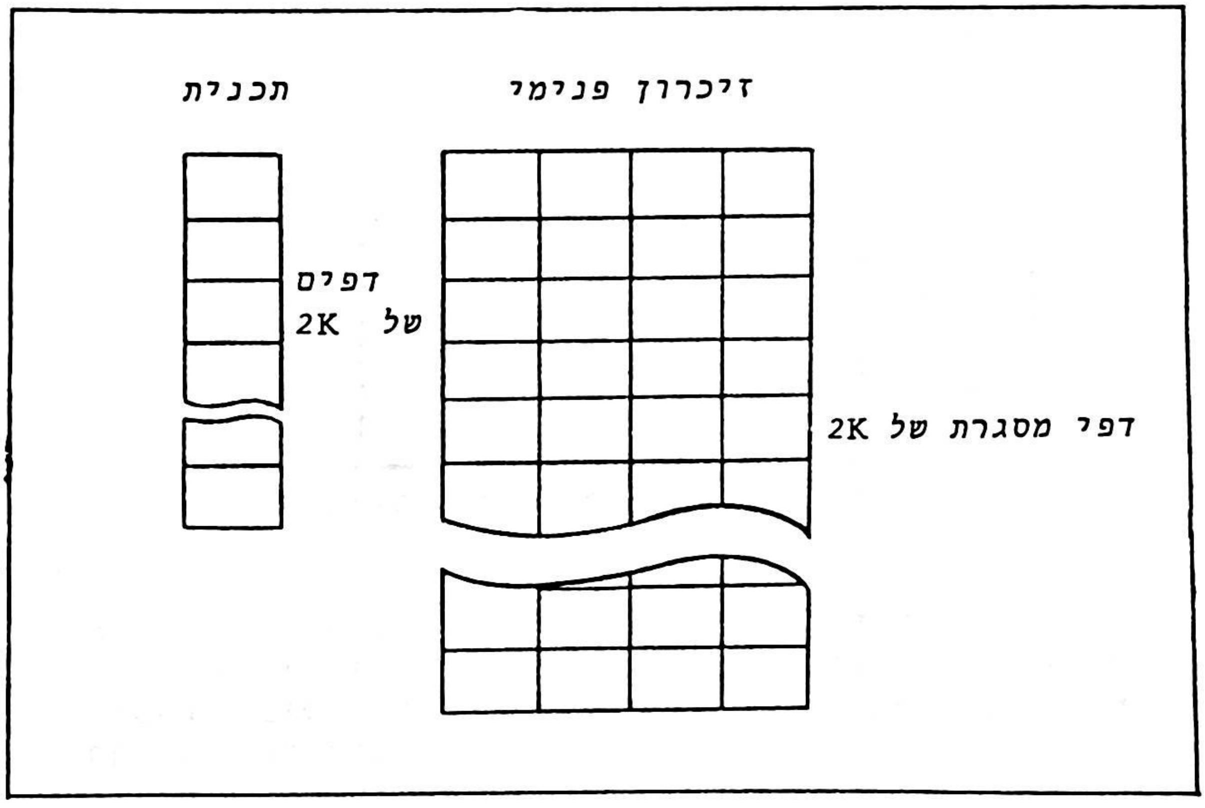
הפתרון לבעיות הנ"ל הינו הזיכרון בפועל (VIRTUAL STORAGE), שהוא טכניקה מיוחדת לניהול הזיכרון. טכניקה זו מתבססת על

השימוש בשלושה מרכיבים: בזיכרון הראשי, באמצעי אחסון לגישה ישירה (דיסקים) ובחמרה ותכנה מיוחדים.

העקרון של זיכרון בפועל הוא שכל הוראה של תכנית כלשהי חייבת להימצא בזיכרון הראשי לקראת הביצוע שלה. אולם אין צורך, שכל החלקים של התכנית יימצאו בזיכרון ביחידת זמן נתונה. הבה נמחיש זאת בדוגמה מחיי היום-יום: נניח שאנו נכנסים למסעדה ומבקשים לאכול ארוחה בת חמש מנות. המלצר יקצה לנו שולחן, נשב סביבו ונזמין את הארוחה בת חמש המנות. המלצר לא יגיש לנו את כל המנות אלא את המנה הראשונה בלבד. לאחר סיום המנה הראשונה, הוא יגיש לשולחננו את המנה השנייה וכו'. בתום הארוחה נפנה את השולחן, כדי לאפשר ללקוח הבא אחרינו לקבל את ארוחתו.

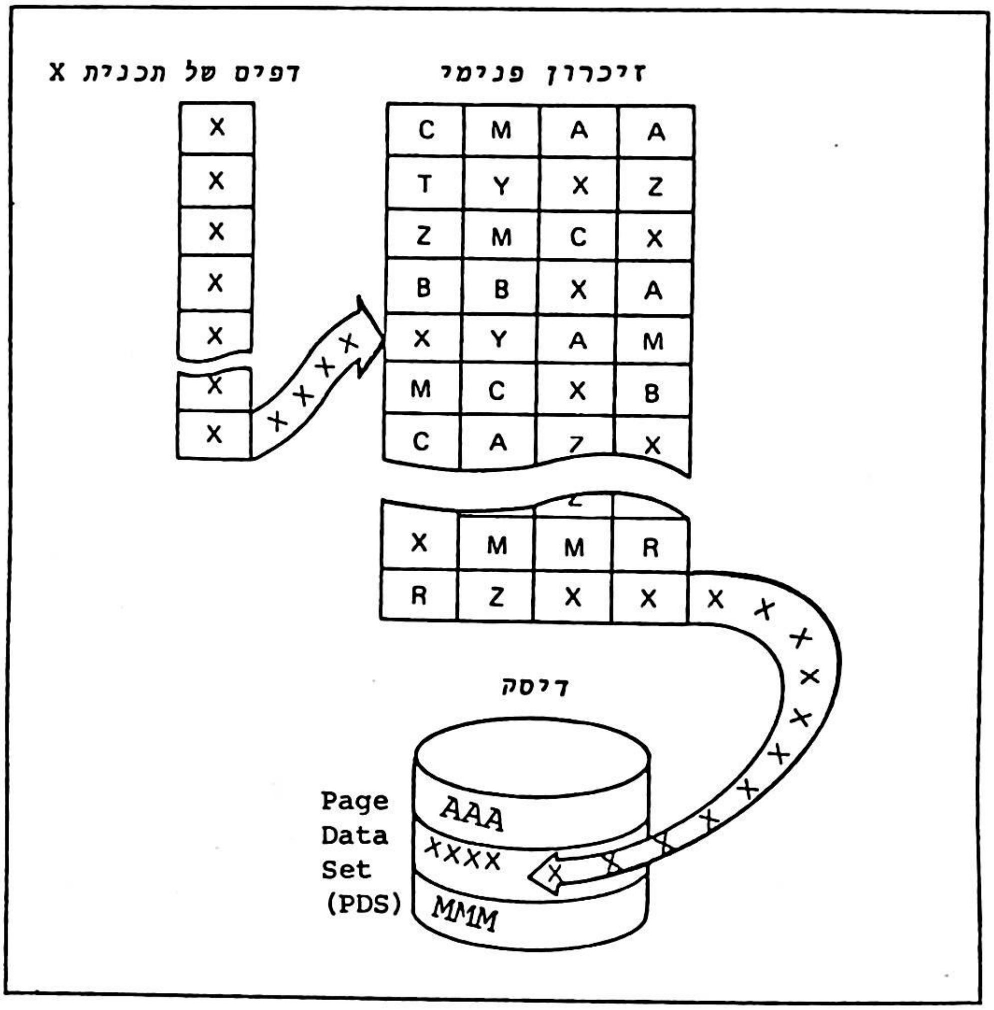
אפשר להשוות את השולחן במסעדה למחיצה בזיכרון, ואת הארוחה בת חמש המנות - לתכנית בת חמישה חלקים או יותר, ואת המלצר - לתכנית הפיקוח של מערכת ההפעלה. אולם קיים הבדל יסודי אחד: בעוד שבזמן האכילה אנו "מחסלים" את המנות המהוות את הארוחה, הרי החלקים של התכנית אינם "מתחסלים" עם ביצועם.

כיצד פועלת שיטת הזיכרון בפועל? בשיטה זו כל תכנית מחולקת לחלקים שווים הנקראים דפים (PAGES), גודל הדפים נקבע ע"י התכונות הפיסיות של המחשב, והוא יכול להשתנות מ-256 תווים במחשבים קטנים ועד 4,096 תווים במחשבים גדולים. גם הזיכרון מחולק לקטעים שווים, הנקראים מסגרות (PAGE FRAMES). בציור 8-7 מוצגים דפים ומסגרות בגודל 2K.



ציור 8-7 - דפים ומסגרות

כפי שהוזכר קודם, בשיטת הזיכרון בפועל יימצאו בזיכרון ביחידת זמן נתונה רק אותם דפים של התכנית, שמתבצעים באותה רגע. דפים אחרים של אותה תכנית ייכתבו על קובץ מיוחד, הנקרא PAGE DATA SET (PDS). מנגנון מיוחד של מערכת ההפעלה דואג לטעינת הדפים הדרושים לביצוע לתוך הזיכרון, ולפינוי הדפים, שאינם בשימוש, לקובץ על הדיסק. תהליך זה נקרא <u>דפדוף</u> (PAGING) והוא מתואר בציור 8-8.



ציור 8-8 - דפדוף

עיון בציור 8-8 מראה, שמסגרות הדפים בזיכרון מכילות דפים של תכניות שונות ורבות. כמו כן ניתן להבין, שהקובץ על הדיסק (PDS) יכיל לא רק את התכנית PROG X אלא כמובן גם את כל התכניות האחרות, שמתבצעות באותה עת.

לסיכום הנושא של זיכרון בפועל, ניתן לציין את הנקודות הבאות:

1. שיטת הזיכרון בפועל מאפשרת ניצול יעיל יותר של הזיכרון, הודות לחלוקתו למסגרות, בהן נמצאים קטעי תכניות (דפים) לביצוע. אין כאן בזבוז של מחיצה בלתי מנוצלת בגלל מימדיה הקטנים (כפי שזה הוצג לגבי תכנית H בתחילת סעיף זה).

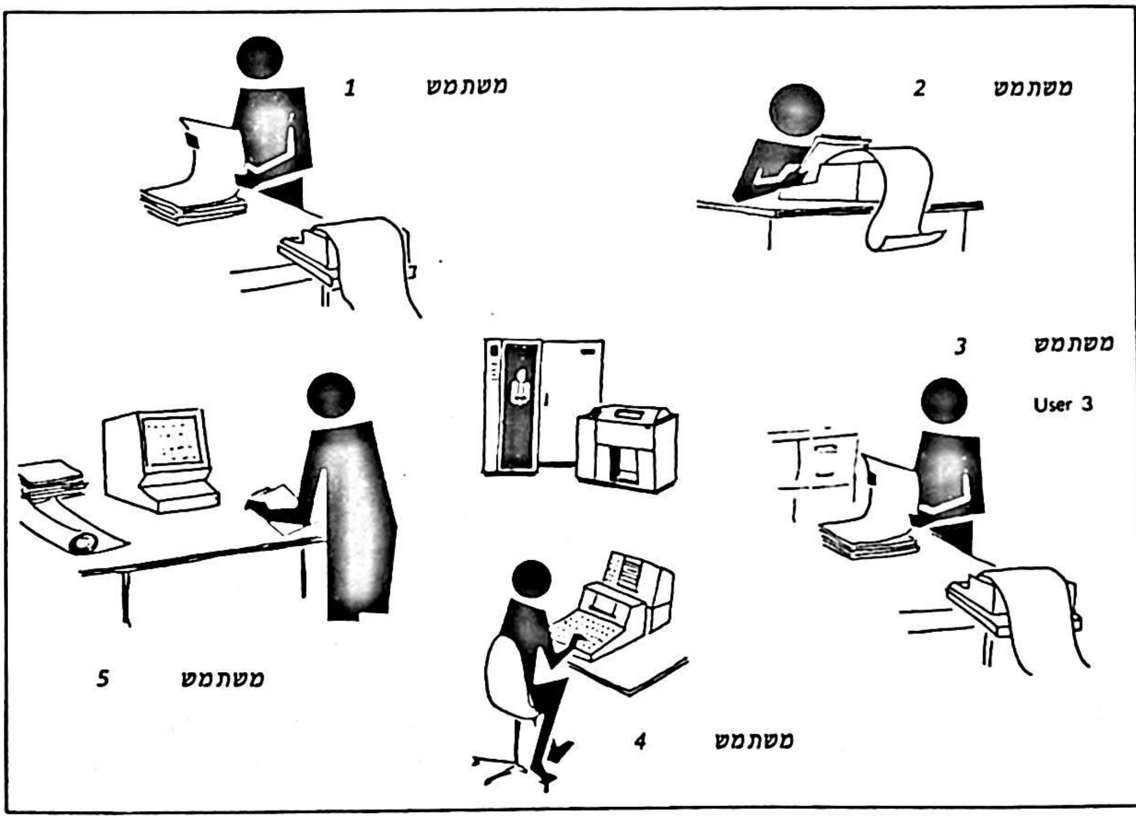
2. ניתן לבצע יותר תכניות ביחידת זמן מסוימת, ופירושו של דבר גם ניצול טוב יותר של היע"מ ויחידות קלט/פלט אחרות.

3. התכניתן משוחרר מחישובי גדלים לגבי התכנית, שהוא כותב. כל תכנית יכולה להיות ארוכה מאד אך למעשה, אורך התכנית מוגבל על ידי שיטת מתן הכתובת בזיכרון. למשל, בסדרת המחשבים הגדולים של חברת יבמ, שיטת הכתובות מאפשרת להשתמש בזיכרון שגודלו התיאורטי יכול להגיע ל-2048MB. זיכרונות פיסיים גדולים הנהוגים כיום מכילים 64MB.

8.7 שיתוף זמן

במונח <u>שיתוף זמן</u> (TIME SHARING) אנו מתכוונים לטכניקה מיוחדת, בה מתאפשר למשתמשים רבים של מערכת מחשב להשתמש באופן סימולטני במערכת. המהירות העצומה, אשר מתבטאת הן בביצוע הוראות והן בהעברת נתונים אל הזיכרון הראשי וממנו, מאפשרת לתת שירותים למספר רב של משתמשי המערכת, אשר עובדים בו-זמנית בשיטת דו-שיח.

שיטה זו מתבססת על עקרונות התיכנות המרובב (ראה סעיף 8.2 לעיל), אך יש בה תוספת חשובה: חלוקת משאבי היע"מ, על-ידי הקצאת "פלח" מזמן היע"מ לכל אחת מתכניות המשתמשים. בצורה זו נדמה לכל משתמש במחשב, שכל מערכת המחשב עומדת לרשותו בלבד (ראה ציור 8-9).



ציור 8-9 - מערכת שיתוף זמן

התנאי לעבוד בשיטה של שיתוף זמן הוא תכנה מיוחדת, אשר מקצה את משאבי המערכת למשתמשים השונים, ומנהלת "רישום" מדויק על

הנעשה במערכת כולה. כמו כן חייבת להיות ספרייה גדולה בעבור תכניות המשתמשים, הקשורים למערכת זו. קשר זה יכול להיות מקומי (LOCAL) או ממרחקים (REMOTE). במקרה של התקשרות מרחוק יש לכלול בשיטה מרכיבים נוספים, הקשורים בעיבוד בתקשורת. נושא זה יידון בפרק 11 של ספר זה.

אופי העבודה בשיתוף זמן הינו הידברותי (CONVERSATIONAL, INTERACTIVE). במהלך השנים פותחו כלים רבים המתאימים לעבודה מעין זו כמו לשפות הידברותיות (APL או BASIC), המאפשרות שיחה כמעט חופשית עם המערכת. שיתוף זמן מהווה כלי יעיל מאד לאנשי מדע ולבעלי מקצועות חופשיים. הוא מהווה שיטה נוחה מאד גם עבור ציבור התכניתנים. פותחו מהדירים מיוחדים לתרגום וניפוי של תכניות בצורה הידברותית (INTERACTIVE DEBUGGING). תכניתן יכול להריץ תכנית, לעצור במקום בו נמצאת שגיאה, לבדוק ולתקן, ואחר-כך להמשיך את ההרצה של התכנית.

ישנן שתי גישות שונות לגבי השימוש בשיתוף זמן. גישה אחת גורסת, שיש להשתמש בשיטה זו בעבור כל משתמשי הארגון. כל מחלקה בארגון מקבלת ציוד מתאים להשתלב בשיטה, ומפתחים בעבורה תכניות מיוחדות. התכניות מותאמות לצרכי המשתמשים, חלקן נושא אופי הידברותי, חלק אחר מתאים לעבודה באצווה.

הגישה השנייה מתבססת על מערכת מחשב חיצונית. הנהלת הארגון יכולה לקבל החלטה, שאין היא מעונינת בכל הדאגות ו"כאבי הראש", הכרוכים בהקמה של יחידת מחשב ובאחזקתה, ועל כן היא תעדיף לפנות אל לשכת שירות, לקבל ממנה ציוד ותכנה מתאימים, אשר יאפשרו עבודה בשיתוף זמן, על המחשב המותקן בלשכת השירות.

ניתן לשלב את שתי הגישות, וכיום קיימים ארגונים רבים במדינת ישראל, שהם בעלים של מחשב קטן או בינוני, המספק את הצרכים היומיומיים והראשוניים של עיבוד נתונים. יחד עם זאת, מחובר הארגון אל לשכת שירות, אשר לה יש מחשב גדול ותכנה מיוחדת, המספקים שירותי שיתוף זמן למשתמשים מסוימים בארגון, שהמחשב הקטן "הביתי" אינו מסוגל לתת.

8.8 מכונה בפועל

צעד נוסף בניצול יעיל יותר של המערכת הינו שיטת המכונה בפועל (VIRTUAL MACHINE או VM), אשר מאחדת בתוכה את התכונות של הזיכרון בפועל ושל שיתוף זמן. בשיטת VM ניתן להתחבר בעזרת מסוף אל מערכת המחשב, ולקבל שירותים המאפשרים למשתמש לעבוד, כאילו הוא קשור למערכת מחשב משלו, עם מערכת הפעלה מיוחדת ויחידות ק/פ השייכות רק לו. זה נעשה בעזרת סימולצייה (הדמיה).

בחברת יבמ, מערכת ה-VM נקראת VM/SP - מהדורה חדשה יותר של VM/370, שהוכרזה באמצע שנות ה-70. זוהי תכנית פיקוח מיוחדת, אשר מנהלת את כל משאבי המערכת, יע"מ, זיכרון, ויחידות ק/פ ויכולה להעמיד אותם לרשות משתמשים רבים. כל משתמש חייב להיות רשום בקטלוג (DIRECTORY) של המערכת. רישום זה מכיל סיסמה, תיאור מערכת המחשב, בו יכול המשתמש לעבוד ופרטים נוספים.

המשתמש מזין למערכת את הסיסמה, ובתשובה הוא מקבל ממנה מקום על דיסק מגנטי, ומערכת הפעלה מיוחדת מאפשרת לו לעבוד, כאילו הוא המפעיל של מערכת מחשב אמיתית.

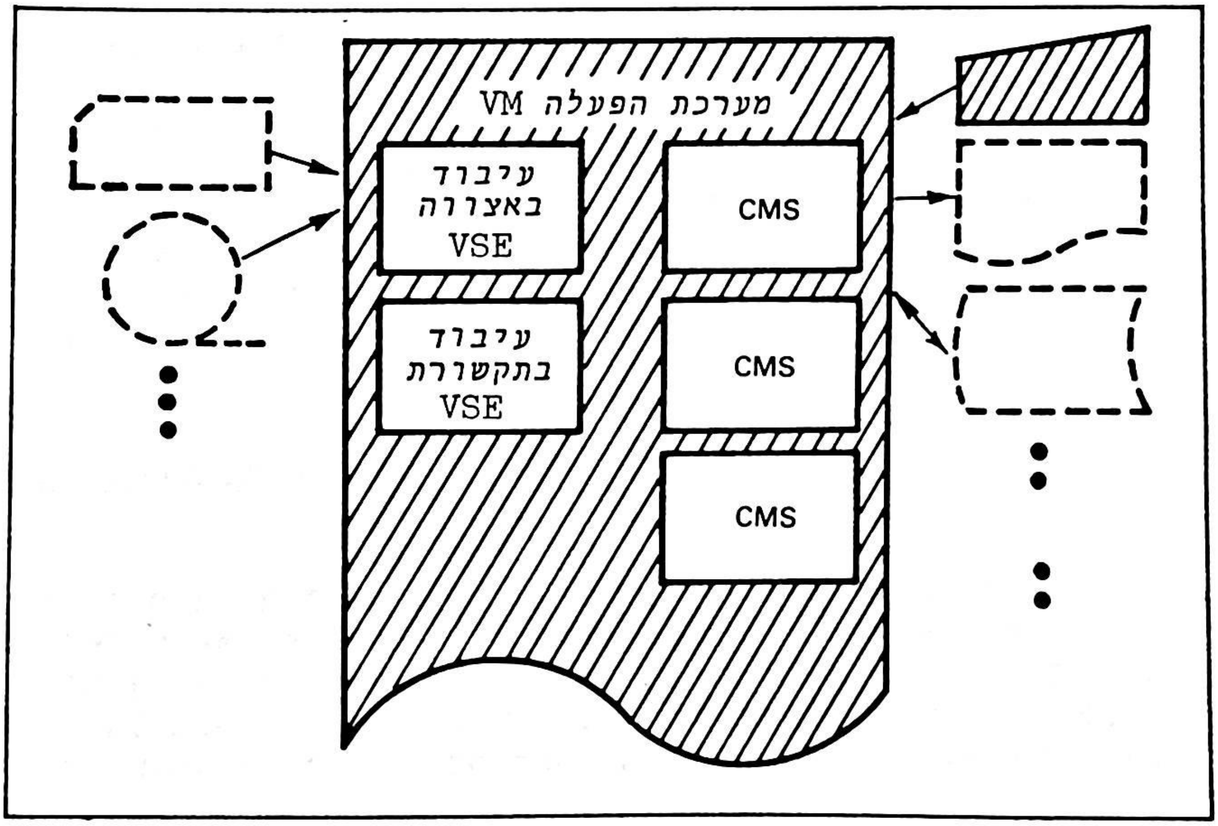
לשיטת VM יתרונות רבים:

- מתן גישה אל המחשב למשתמשים רבים, כאשר לרשות כל אחד עומדת לכאורה מערכת מחשב שלימה, עם צירוף יחידות ק/פ הדרושות לו ועם זיכרון כמעט בלתי מוגבל.

- אפשרות לעבודה הידברותית, הנוחה מאד לאנשי מקצוע טכני, מינהלי או ניהולי, הרוצים מידע מועט אך מידי.

- למרות שקיים רק מחשב אחד, ניתן "להריץ" בו תכניות תחת מערכות הפעלה שונות.

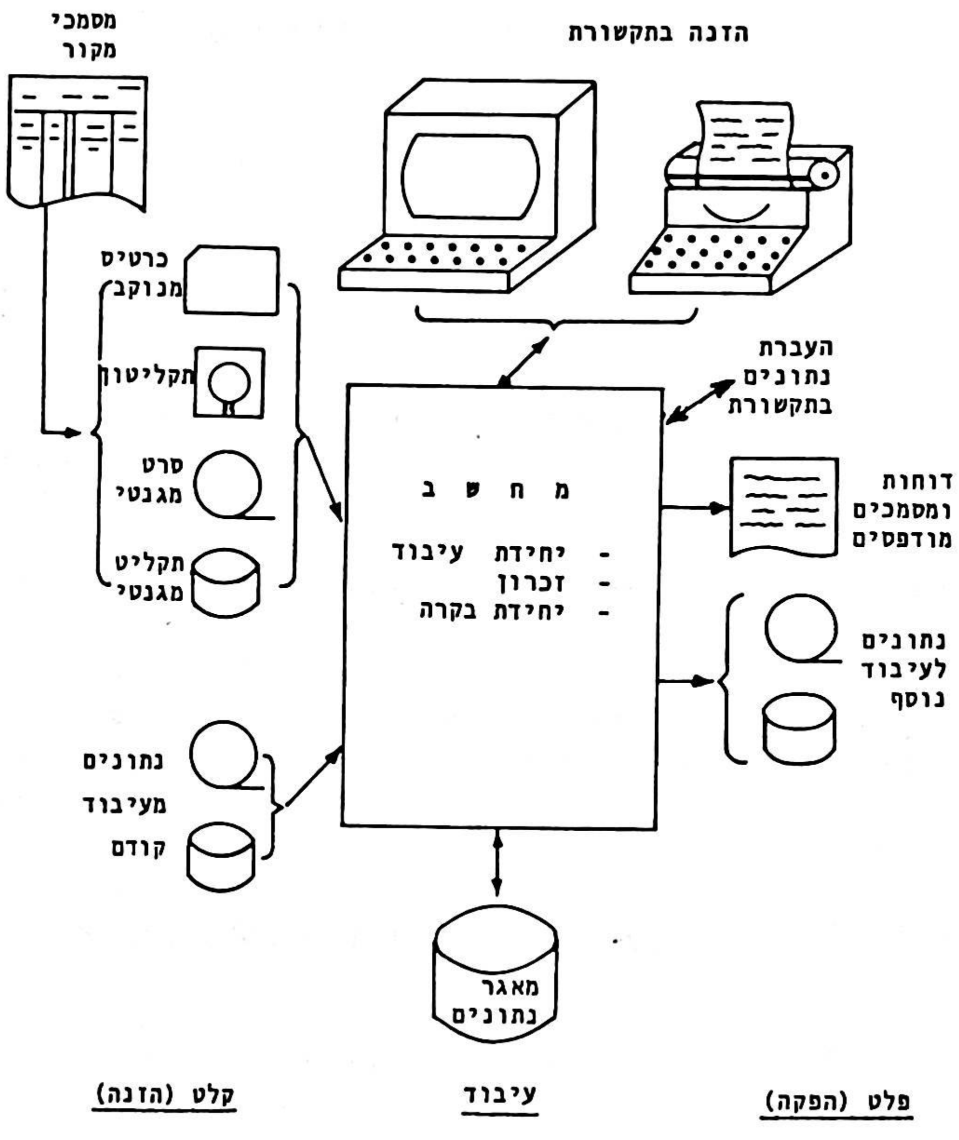
- ארגון מסוים רוצה להחליף את המחשב. תחת VM/SP ניתן להמשיך את העבודה במתכונת הישנה, ולהכין את כל התכניות במערכת החדשה.



ציור 8-10 - מערכות הפעלה שונות תחת VM

ציור 10-8 ממחיש את יתרונות ה-VM. בדוגמה שלפנינו מוצגות 5 מערכות מחשב שונות. שתים מהן - VSE, פועלות באצווה, ושלש מערכות CMS אשר מופעלות ע"י משתמשים שונים.

תפיסת העבודה של VM נתמכה על ידי מספר חברות מחשבים בעולם. חברת NCR לדוגמה, הכריזה על שירותי VM בסדרת ה-CRITERION שלה בשנת 1976. מחשבי ה-V-8560 אפשרו למכונות קטנות יותר של NCR (CENTURY 300 למשל) לעבור עם התכניות והקבצים אל המערכת הגדולה. במחשבי DATA GENERAL מופעלת גם כן מערכת הפעלה VM הנקראת VMS.



מערך קליטה ועיבוד נתונים

פרק 8 - שאלות

1. הסבר בקצרה מדוע שיטת העבודה במחשבים לא היתה יעילה בשנות ה-60.

2. הסבר את המושגים "תכנות מרובב", "עיבוד מרובב".

3. מהי תכנית המשגוח?

4. מנה את המרכיבים העיקריים של מערכת ההפעלה.

5. בעת הביצוע של תכנית מחשב מופיעה תקלה. איזה חלק של מערכת ההפעלה יטפל בה?

6. כיצד יכול המפעיל לדעת אם נתגלתה תקלה באחת התכניות?

7. למערכת מחשב מסוימת יש רק מדפסת אחת. ביחידת זמן מסוימת פועלות 5 תכניות לפי שיטת התכנות המרובב, אשר רוצות להשתמש במדפסת. מה יקרה לדעתך?

8. הסבר, בכמה משפטים, מהו זיכרון בפועל.

9. מה היתרון בשיטת הזיכרון בפועל?

10. הסבר את המושגים LOCAL, REMOTE.

11. מהם היתרונות של מכונה בפועל?

פרק 9 – ארגון נתונים וניהול קבצים

בן-דודי שמואל, שהוא חבר קיבוץ, סיפר לי, שאסיפת החברים הטילה עליו לקבל את האחריות למחסן הכלים במשק. הבעיה, כפי שהוא תיאר אותה, היא שהמחסנאי הקודם עזב את המשק, והשאיר אי-סדר בניהול המלאי. שמואל, שעבד כל השנים ברפת, לא התמצא די הצורך בנעשה במחסן, ולא הכיר את סדר הכלים, הברגים והמסמרים. הוא לא ידע כיצד לסדר את המחסן, כדי שזה ישרת אותו ואת הקיבוץ בצורה הטובה ביותר. טלפנתי אליו והבטחתי לבוא למשק כעבור שבועיים, כדי לעזור לו במידת האפשר.

כאשר הגעתי לקיבוץ, מצאתי את שמואל במחסן, שהיה מבריק ומסודר. "כיצד הצלחת להתגבר כה מהר על אי-הסדר הנורא?" שאלתיו, "בתחילה זה היה קשה, אך לאחר שמצאתי את השיטה, הכל הפך להיות קל! ריכזתי את הכלים בפינה אחת של המחסן, שקי מלט וסיד שמתי קרוב לדלת, כדי שיהיה קל להוציא אותם, פחי צבע שמתי על המדף העליון, והזמנתי קופסאות מיוחדות למסמרים ולברגים השונים, כדי שיהיה נוח למיין אותם לפי סוג. כלי העבודה יהיו על המדף האמצעי, כך שבקלות רבה יהיה ניתן לגשת אליהם...". שמואל המשיך בסיפורו, אך אני כבר לא הקשבתי לדבריו. חשבתי מה רב המרחק בין מחסן הקיבוץ לבין מחשב בבנק בעיר, ובכל זאת כמה הם דומים בשיטה...

נתונים הם חומר הגלם של תהליך העיבוד במערכות ממוחשבות. הנתונים זקוקים לארגון נכון ויעיל לפחות באותה המידה בה מסודרים החומרים במחסן הקיבוץ. למעשה, הטיפול בנתונים מורכב ומסובך עוד יותר, וקיימות מספר שיטות לארגון ולעיבוד הנתונים הללו. היום, יותר מאשר בעבר, כל אדם הינו צרכן או משתמש פוטנציאלי של מערכת מחשב. מחשבים נעשים זולים יותר ויותר, ותופסים את מקומם לא רק במוסדות ובמפעלים, אלא גם במשרד הקטן או בבית.

תשומת הלב מתמקדת עתה בתהליך עיבוד הנתונים: מה לעבד וכיצד לעבד בעזרת מחשב, היכן מתחילה פעולת עיבוד הנתונים, ובאיזה מקום היא מסתיימת. לשאלות אלו רוצים אנו למצוא תשובות בפרק זה העוסק בנתונים, בארגונם ובעיבודם.

9.1 ניהול נתונים

הנתונים הם אחד מהמשאבים החשובים של מערכת המחשב; הם משמשים

כחומר הגלם בתהליך המרכזי של המערכת - עיבוד הנתונים (DATA PROCESSING).

הנתונים נמצאים בזיכרון הראשי בזמן העיבוד, אולם היכן נשמור אותם בין עיבוד לעיבוד? ניקח לדוגמה את קובץ העובדים במפעל, שמשתמשים בו אחת לחודש לחישוב המשכורת. בזמן החישוב של המשכורת ובזמן הדפסת התלושים, הרשומה של כל עובד חייבת להימצא בזיכרון המרכזי. אך לאחר הדפסת תלוש המשכורת אין צורך ברשומה זו למשך חודש ימים, עד לחישוב המשכורת בחודש הבא (אלא אם רוצים לעדכן את הרשומה, אך על כך ידובר מאוחר יותר).

בפרק 4 של ספר זה תוארו אמצעי הקלט, כגון קוראת כרטיסים וכונן סרט מגנטי, שבעזרתם מזינים את הנתונים למערכת. עתה נדון באמצעי האחסון העיקריים של מערכת המחשב: הסרט המגנטי והדיסק המגנטי. במערכות מחשב קטנות, במיקרו-מחשבים ובמחשבים האישיים, משמש התקליטון כאמצעי אחסון עיקרי. במחשבים ביתיים זעירים משתמשים לאחסנה בקלטות של רשמקול.

חשיבות רבה נודעת לסדר, בו מאוחסנות הרשומות, האם הרשומות תסודרנה לפי אי-בי או לפי סדר אחר? האם אורך הרשומות בקובץ, קבוע או משתנה? האם הקובץ יאוחסן על סרט מגנטי או על דיסק מגנטי? בשאלות אלו ואחרות עוסקים בארגון הנתונים (DATA ORGANIZATION). זהו תחום הכולל גם תכניות שונות, שתפקידן ליצור, לטפל ולתחזק נתונים על אמצעי אחסון מגנטיים.

אך קיימת בעיה נוספת: כיצד להעביר את הנתונים מאמצעי האחסון החיצוניים הללו אל הזיכרון המרכזי? כיצד תאותרנה הרשומות בתוך הקובץ? האם נקרא לזיכרון רשומה אחר רשומה או ניתן לקרוא לרשומה בודדת כלשהי? האם אפשר לעדכן רשומה בזיכרון ואחר-כך לכתוב אותה מחדש במקום הרשומה הישנה, או שיש לחפש בעבורה מקום חדש על אמצעי האחסון? בעיות אלו נפתרות בעזרת שגרות שונות, הנקראות שיטות גישה (ACCESS METHODS). שגרות אלה נכתבות על-פי-רב על ידי יצרני המחשבים או על-ידי יוצרי מערכות ההפעלה.

השם הכולל, שניתן לתכניות ולשגרות, המטפלות בארגון הנתונים ובשיטות הגישה אליהן, הינו ניהול נתונים (DATA MANAGEMENT).

לפני שנדון בפרטי שיטות האחסון, נציין מספר עובדות חשובות:

- כמות הנתונים הדורשים עיבוד, הולכת וגדלה מיום ליום;

- הנתונים עוברים בדרך כלל עיבוד חוזר במערכת המחשב בפרקי זמן שונים;

- עיבוד נתונים יעיל מותנה בארגון נכון ובאחסנה מתאימה.

קיים קשר בין השיטה, בה מאורגנים הנתונים על אמצעי האחסון

המגנטיים, לבין שיטת הגישה אליהן ואופן עיבודם. הסעיפים הבאים של פרק זה כוללים תיאור של סוגי ארגון שונים וטכניקות עיבוד, המתייחסות אליהם.

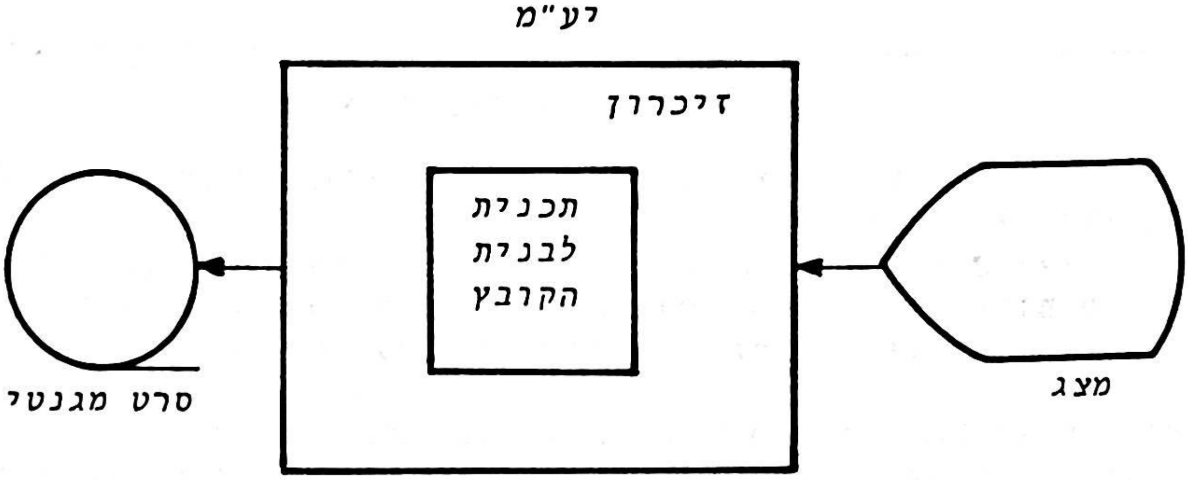
9.2 ארגון נתונים בסרטים מגנטיים

הסרט המגנטי הינו אמצעי אחסנה זול ונפוץ מאד, אולי הנפוץ ביותר למטרת אחסון נתונים.

ניתן לכתוב את הרשומות של קובץ כלשהו על סרט מגנטי בתבניות שונות. ניקח לדוגמה את המקרה הבא: מורה בבית-ספר ערך רשימות על תלמידיו במשך כל שנת הלימודים. הרשימות רוכזו בטבלה כזו:

שם התלמיד	מקצוע	ציון שליש א׳	ציון שליש ב׳
----------	-----	-----	-----
----------	-----	-----	-----

לאחר שהגיע מחשב לבית הספר, התעניין המורה כיצד ניתן לשמור את רשימותיו כקובץ תלמידים, כדי שיוכל להשתמש בקובץ בעתיד למעקב, למתן ציונים ועוד. הוא החליט להזין את הנתונים למחשב דרך מצג, כדי שייכתבו ויאוחסנו על סרט מגנטי (ראה ציור 9-1).



ציור 9-1 - בנית קובץ על סרט מגנטי

הקובץ, אשר ייבנה על הסרט המגנטי, יכיל רשומות באורך קבוע. תיאור סכימטי ניתן בציור 9-2.



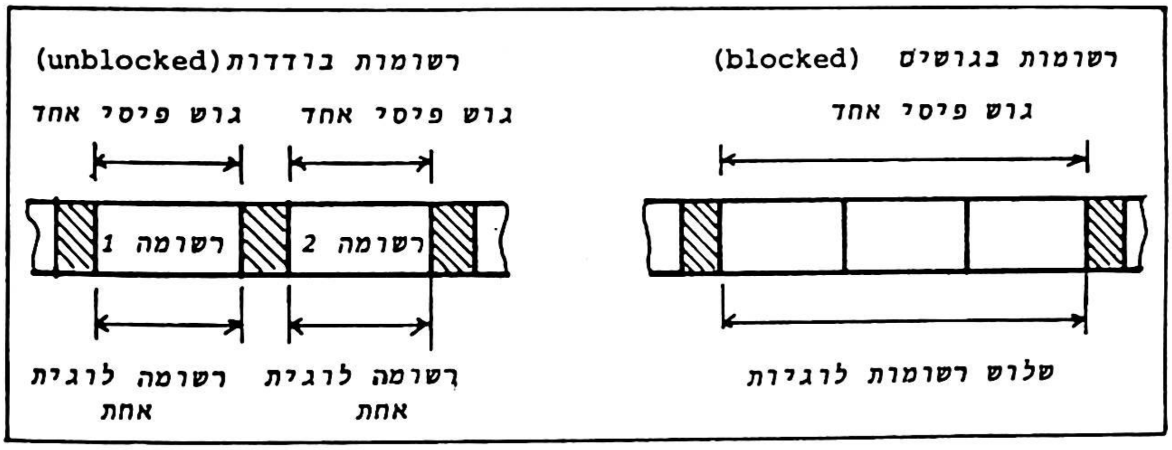
ציור 2-9 - רשומות באורך קבוע

לאחר כל רשומה נמצא קטע "ריק" על הסרט, הנקרא **מרווח** (INTER RECORD GAP - GAP). מרווח זה משמש לשתי מטרות:

1. להפריד בין הרשומות או בין גושי רשומות.

2. להוות מרחק תאוצה ועצירה לפני ואחרי קריאה או כתיבה של גוש נתונים.

המרווח נבנה באופן אוטומטי, לאחר סיום הכתיבה של קבוצת נתונים. גודלו בדרך כלל כחצי אינטש. בצפיפות הנתונים על סרטים מגנטיים של 1,600-6,250 תווים לאינטש "מתבזבז" מקום של 800 תווים, ובצפיפות של 6250, הבזבוז הוא של 3125 תווים. כדי להתגבר על תופעה זו נכתבו תכניות מיוחדות, המאפשרות ליצור **גושים**, אשר מכילים מספר רשומות (ראה ציור 3-9).



ציור 3-9 - רשומות בגושים

החיסכון כאן משמעותי ביותר: נניח שיש קובץ בן 15,000 רשומות. אם הקובץ ייבנה לפי תבנית מס' 1 (ציור 4-9) לא ינוצלו בסרט 12,000,000 תווים!. בקובץ הבנוי לפי תבנית מס' 2 (ציור 5-9) יתפסו המרווחים רק 4,000,000 תווים, והחיסכון במקום משמעותי ביותר. כמו כן יש לזכור את החיסכון בזמן הקריאה.

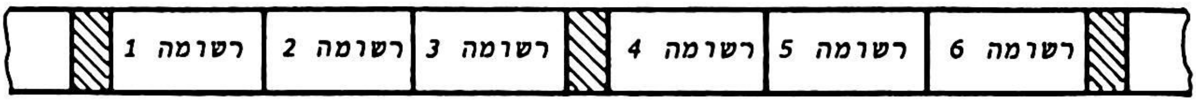
עתה נוכל להגדיר גוש: **גוש** (BLOCK) הינו קטע של קובץ, הנמצא בין שני מרווחים בזמן של פעולות קלט/פלט. פעולת הקריאה או הכתיבה מתבצעת על גוש נתונים אחד. גוש כזה יכול להכיל רשומה אחת או יותר באורכים שווים או שונים. להלן תיאור קצר של התבניות השכיחות של הרשומות והגושים:

1. **תבנית 1** - רשומה אחת לגוש. הרשומות באורך קבוע (UNBLOCKED, FIXED LENGTH) - ראה ציור 4-9.



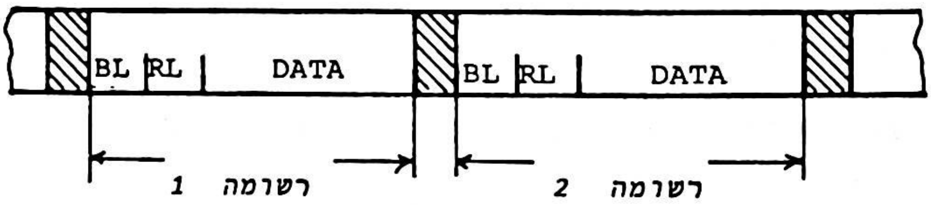
ציור 9-4 - תבנית מס' 1

2. <u>תבנית 2</u> - מספר רשומות בגוש וכל רשומה באורך קבוע (BLOCKED, FIXED).



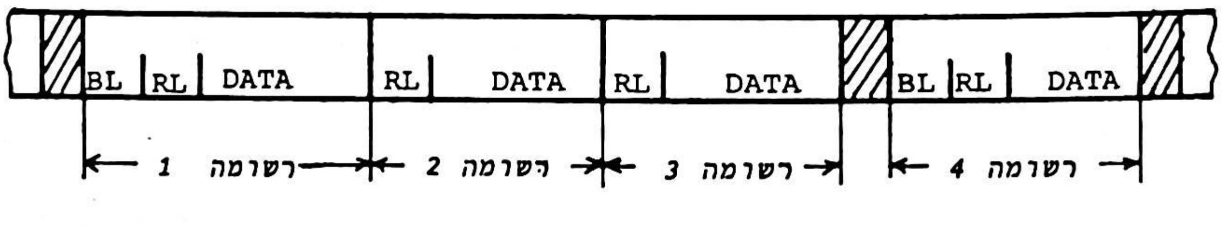
ציור 9-5 - תבנית מס' 2

3. <u>תבנית 3</u> - רשומה אחת באורך משתנה בכל גוש (UNBLOCKED, VARIABLE LENGTH). BL ו-RL מציינים את <u>אורך הבלוק</u> (BLOCK LENGTH) ואת <u>אורך הרשומה</u> (RECORD LENGTH) בהתאמה.



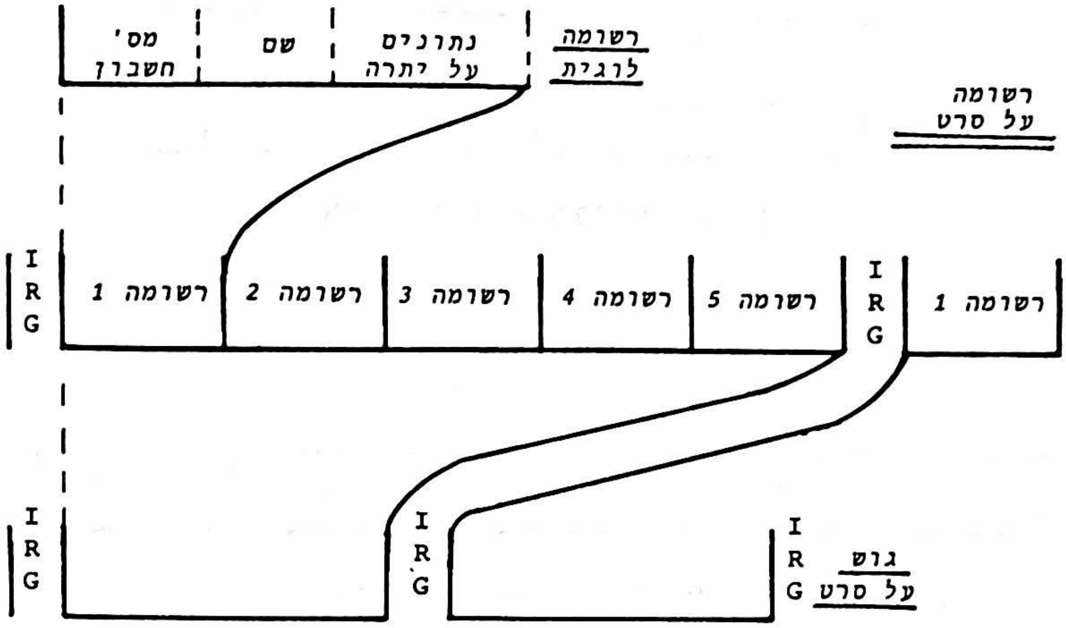
ציור 9-6 - תבנית מס' 3

4. <u>תבנית 4</u> - מספר רשומות בעלות אורכים שונים בגוש אחד (BLOCKED, VARIABLE LENGTH). BL ו-RL כמו בסעיף 3.



ציור 9-7 - תבנית מס' 4

בתבניות 2, 4 נוהגים גם לציין מהו ה-<u>BLOCKING FACTOR</u>, ז"א מהו מספר הרשומות, הנמצאות בגוש אחד. גורם זה חשוב בתכנון העבודה לשם יעול העיבוד.



ציור 9-8 - שדה, רשומה, גוש

9.3 ארגון נתונים בדיסק מגנטי

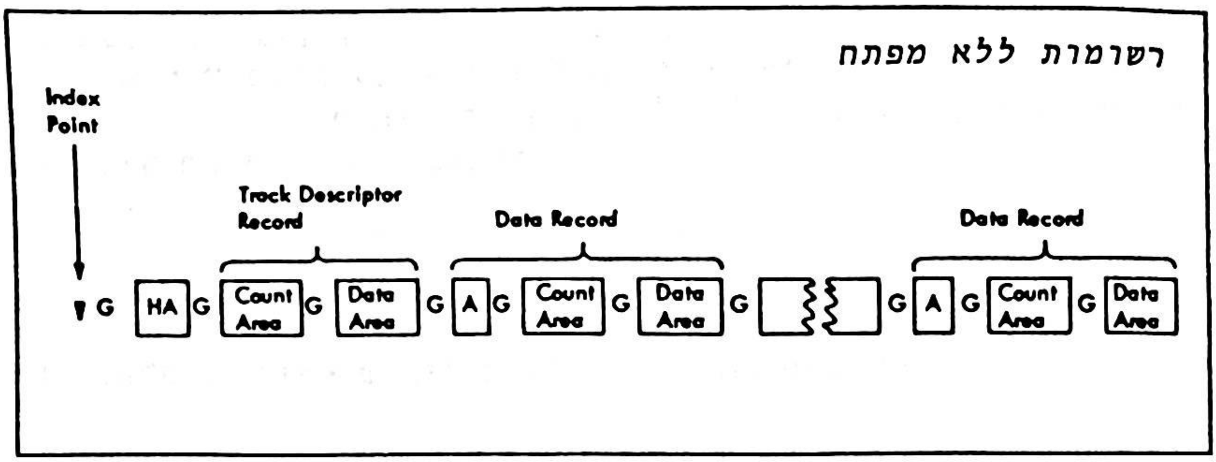
בפרק אחר צוין, שהדיסק המגנטי הינו אמצעי אחסון חיצוני המאפשר, בגלל תכונותיו הפיסיות, גישה ישירה אל כל רשומה בודדת בקבצים השמורים עליו. הנתונים רשומים על המסילות הקונצנטריות באחת משתי צורות, כפי שמוצג בציור 9-9.

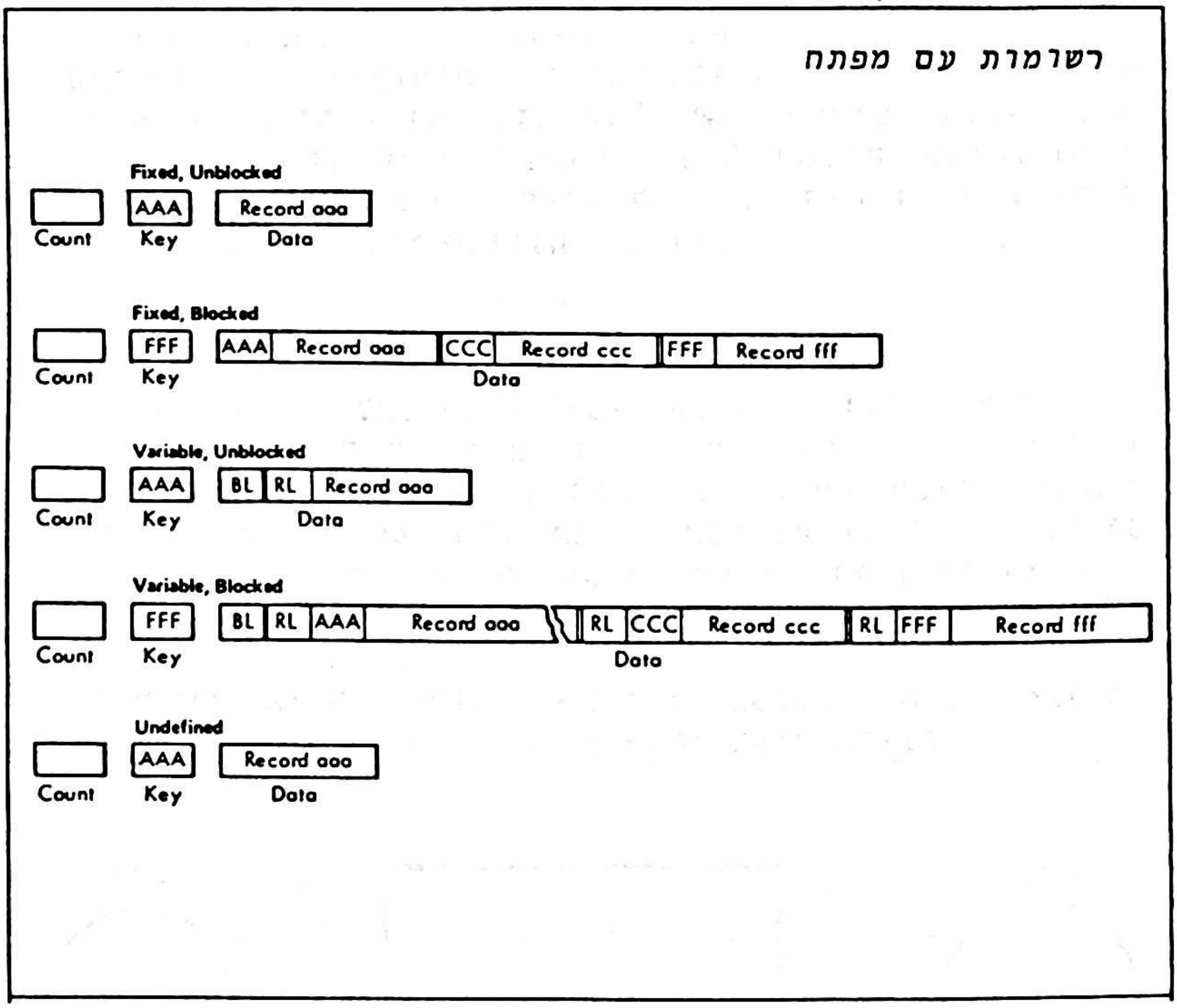
- רשומות בעלות מפתח.
- רשומות ללא מפתח.

בשני סוגי הרשומות ניתן להבחין בשדה COUNT (מונה), המכיל מידע לפיו ניתן לאתר את הרשומות (מספר הגליל, מספר המסילה, ומספר הרשומה בתוך המסילה). שדה המפתח הינו בד"כ שדה מספרי, אשר מאפשר לאתר את הרשומות בקלות, כפי שמאתרים חבר לפי כתובת שמסר לנו.

תבניות הרשומות על דיסק מגנטי

כל ארבע התבניות של הרשומות, אשר הוזכרו בקשר לסרטים מגנטיים, קיימות גם על דיסק מגנטי. קיים גם סוג נוסף - והוא הרשומה המפוצלת (SPANNED RECORD). זוהי רשומה גדולה, שניתן לכתוב אותה כך, שחלקה נמצא על מסילה אחת וחלקה האחר על מסילה אחרת. כל מערכות ההפעלה תומכות בעיבוד קבצים המכילים סוג כזה של רשומות.





ציור 9-9 - תבנית רשומה בדיסק

9.4 עיבוד קבצים לפי שיטת הגישה הסדרתית

<u>שיטת הגישה הסדרתית</u> (SEQUENTIAL ACCESS METHOD - SAM) היא השיטה הפשוטה ביותר של ארגון נתונים. בקובץ המאורגן לפי שיטה זו נמצאות הרשומות זו אחר זו לפי סדר, שנקבע מראש. סדר זה מתבסס על אחד (או יותר) משדות הרשומה. על-פי-רב השדה הזה הוא

שדה מספרי, כגון מספר תעודת זהות או מספר קטלוגי. שדה כזה נקרא גם שדה בקרה (CONTROL FIELD) או מפתח (KEY). כאן המקום לציין, שקובצי כרטיסים וקבצים על סרטים בעבור מדפסות מאורגנים תמיד בארגון סדרתי.

9.4.1 קובץ סדרתי על סרט מגנטי

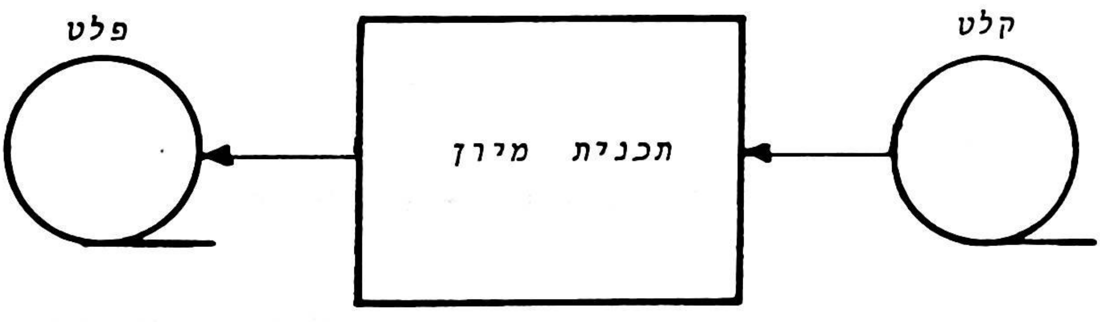
ניתן לחלק את נושא עיבוד הקבצים לשלושה חלקים:

1. יצירת הקובץ (CREATION)

בפעם הראשונה, כאשר יוצרים את הקובץ הסדרתי, קובעים את סדר המיון של הרשומות. למשל, קובץ עובדים ימויין לפי שדה של מספר תעודת זהות וסדר עולה של מספרים. הסדר, שנקבע בפעם הראשונה, חייב להישמר, וכל רשומה חדשה, שתתוסף לקובץ, תיכנס במקום המתאים לה. במקרים רבים נוצר סדר המיון בפעולת עיבוד מכוונת - מיון.

2. מיון הקובץ (SORT)

המיון הינו פעולה, בה מחליטים לסדר את הרשומות של הקובץ לפי קריטריונים מסוימים, הנדרשים באותה בעת. למשל קובץ עובדים במפעל ניתן למיון לא רק לפי מספר תעודת זהות אלא גם לפי גובה המשכורת, הגיל, הכתובת וכו'. כאשר נכתב הקובץ על סרט מגנטי, תתבצע פעולת המיון בדרך הבאה: על כונן הקלט יורכב הקובץ הישן, תכנית המיון תמיין את הרשומות, אשר תיכתבנה לפי הסדר החדש, על סרט אחר, על כונן הפלט (ראה ציור 9-10). יש לציין, שפעולת המיון אינה משנה את תוכן הרשומות אלא רק הסדר שלהן בתוך הקובץ.



ציור 9-10 - מיון קובץ בסרטים מגנטיים

3. עדכון הקובץ (UPDATE)

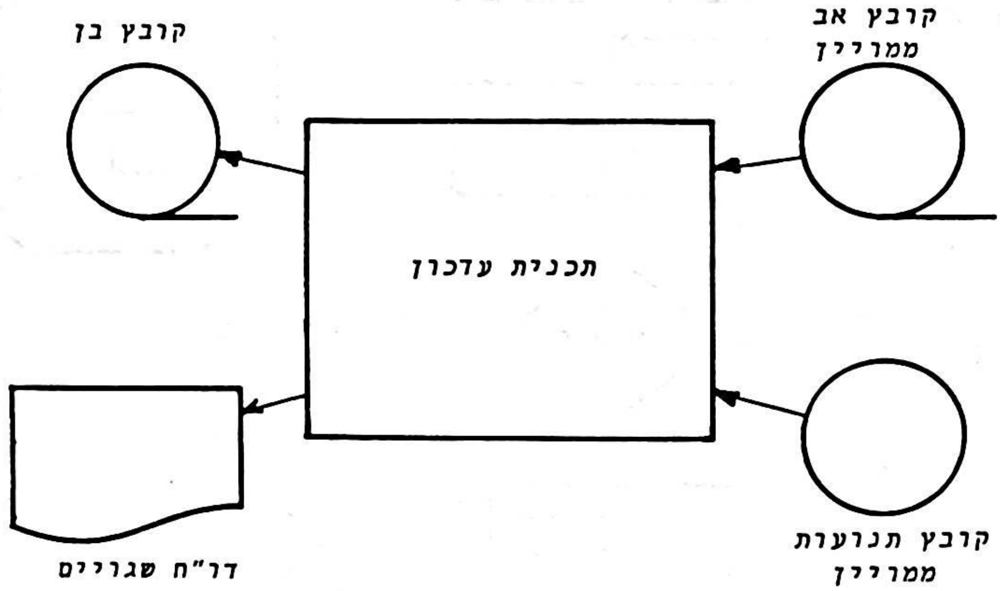
העדכון מתיחס בדרך כלל לשלושה סוגי פעולה שונים:

א. הוספת רשומה חדשה לקובץ;

ב. עדכון שדה או כמה שדות בתוך הרשומה;

ג. ביטול רשומה מתוך הקובץ.

נתאר כאן עדכון קובץ סדרתי בעזרת דוגמה של קובץ "קופת חסכון". העיבוד מתואר בציורים 9-11, 9-12 (תרשימים) ו-9-13 (טבלאות מספריות).



ציור 9-11 - עדכון קובץ בתרשים זרימה

בעיבוד נבחין <u>בשני קבצי קלט</u>:

1. <u>קובץ אב</u> (MASTER FILE) - קובץ אשר מכיל את <u>כל</u> הרשומות של החוסכים בקופה;

2. <u>קובץ תנועות</u> (TRANSACTION FILE) - קובץ זה מכיל רק את הרשומות שחלו בהן שינויים מאז העדכון האחרון.

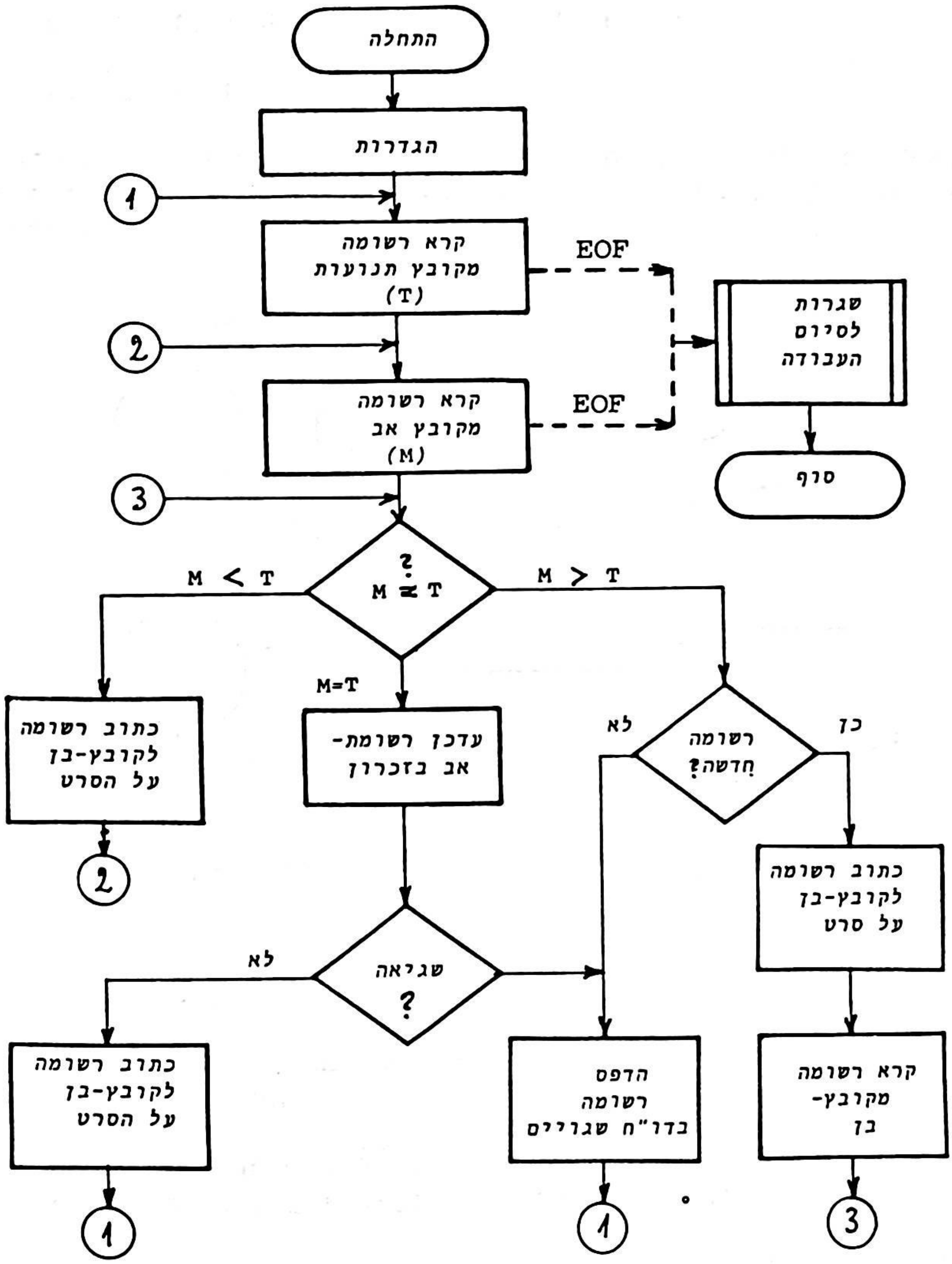
שני הקבצים <u>חייבים</u> להיות ממויינים לפי אותו שדה, ובמקרה זה - "מספר חוסך".

בנוסף יהיו שני <u>קבצי פלט</u>:

1. <u>קובץ בן</u>, שהוא למעשה קובץ אב חדש, המכיל את כל העדכונים.

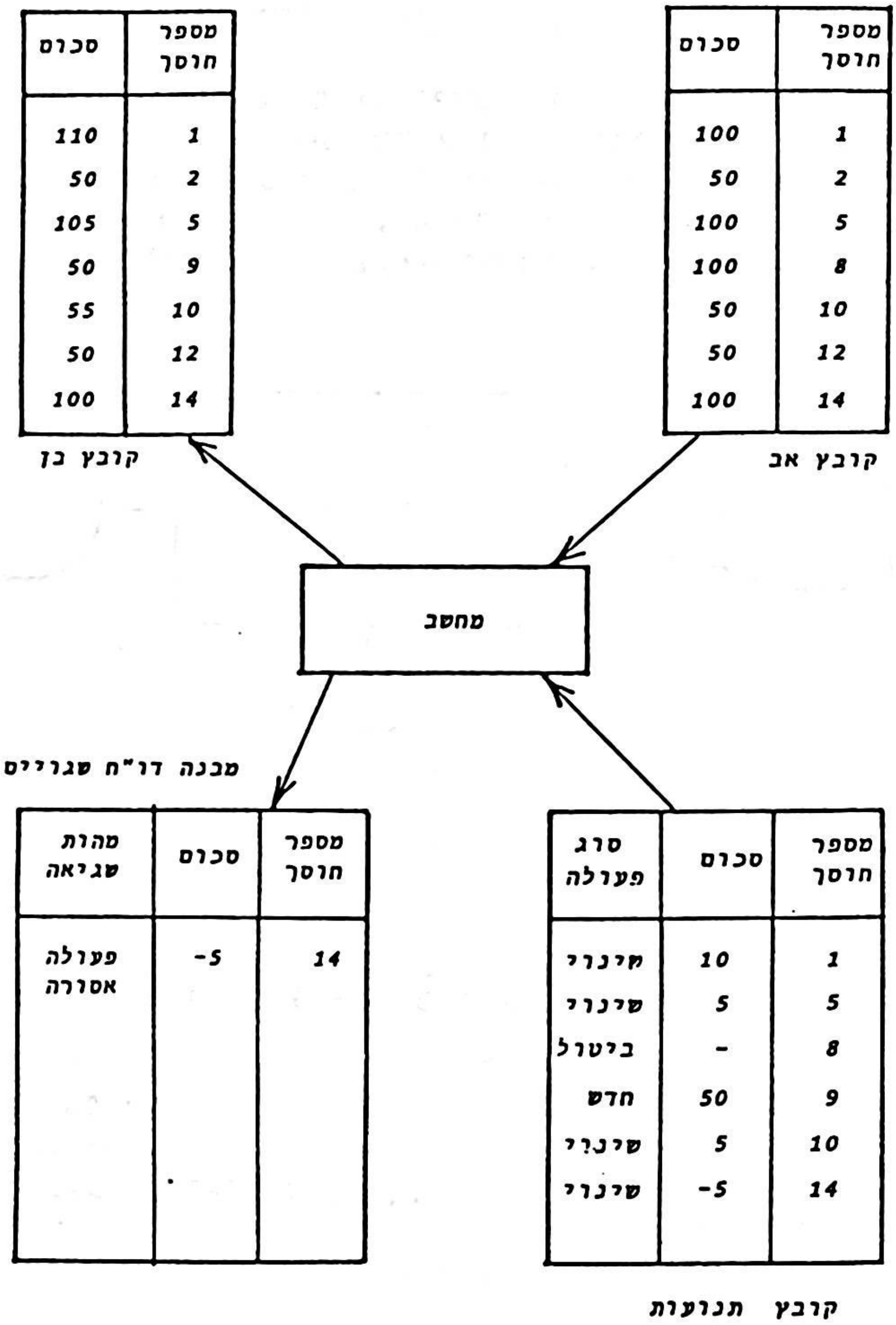
2. קובץ "<u>דו"ח שגויים</u>", שיכיל את כל הרשומות בהן נתגלתה שגיאה בתוך כדי עדכון.

שלבי תכנית העדכון מתוארים בתרשים 9-12.



ציור 9-12 - תרשים תהליך העדכון

ניתן לעקוב אחר ביצוע התכנית לפי הטבלאות שבציור 9-13. כל טבלה מייצגת קובץ מסוים בעיבוד זה, כאשר בטבלה מופיעים רק השדות הרלבנטיים לכאן.



ציור 13-9 - הצגת פעולות עדכון

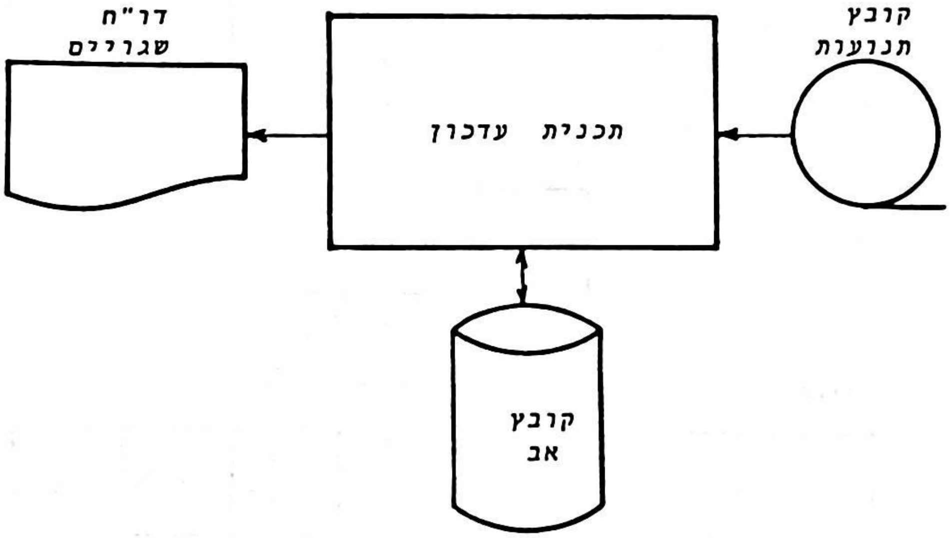
הקורא יכול לעקוב בעצמו אחרי שלבי העיבוד השונים, לפי הטבלאות. עדכון קבצים יסתיים, כאשר מגיעה התכנית לסוף הקובץ בשני קובצי הקלט. יש להדגיש כאן נקודות חשובות:

1. בתוך כדי העיבוד הועתקו כל הרשומות של קובץ האב לקובץ הבן, בין אם היה בהן שינוי ובין אם לאו. צורה זו של עדכון אופיינית לכל הקבצים המאוחסנים בסרטים מגנטיים.

2. בעדכון קבצים על סרטים מגנטיים דרוש תמיד סרט חדש לקובץ הפלט, בנוסף לסרט המקורי, שבו נמצא קובץ הקלט. אם הקובץ משתרע על יותר מגלגל אחד של סרט - מספר הגלגלים החדשים זהה בדרך-כלל למספר הגלגלים הישנים.

9.4.2 קובץ סדרתי על דיסק מגנטי

קבצים סדרתיים מאוחסנים גם על דיסקים מגנטיים, ותהליך העיבוד שלהם אינו שונה באופן מהותי מתהליך העיבוד של קבצים סדרתיים על סרטים מגנטיים. בקבצים על דיסקים אין צורך להעתיק את הקובץ מחדש. אם אין רוצים בכך, אפשר לעדכן את הרשומה ולהחזירה למקומה הקודם. תהליך העדכון בדיסקים מוצג בציור 14-9.



ציור 14-9 - עדכון קובץ סדרתי

שלבי העדכון הם :

1. רשומת תנועה נקראת לזיכרון;
2. רשומת אב בעלת אותו מספר מזהה נקראת לזיכרון;
3. רשומת אב מתעדכנת לפי רשומת תנועה;
4. רשומת אב מעודכנת נכתבת בחזרה למקומה המקורי (אלא אם כן התכנית דורשת לכתבה במקום אחר בגלל סיבות שונות, או שחל שינוי באורך שלה).

9.5 ארגון ועיבוד קבצים על דיסקים מגנטיים

שיטת העיבוד הסדרתי בדיסק מגנטי הוצגה בסעיף קודם. עתה נדון באפשרויות עיבוד אחרות. קיימים מספר סוגים של אמצעי אחסנה, אשר מאפשרים גישה ישירה אל רשומה יחידה מתוך הקובץ. שמם הכולל באנגלית: DASD (DIRECT ACCESS STORAGE DEVICE). האמצעי

הנפוץ מבין כולם הינו הדיסק המגנטי. אמצעי אחסון נפוץ אחר הוא התקליטון.

כדי להמחיש את היתרונות של גישה ישירה, יחשוב הקורא על מקרה, בו מחפשים מלה מסוימת במילון גדול מאד. ניתן לעשות זאת בשתי שיטות:

1. <u>שיטת הגישה הסדרתית</u> - לפיה קוראים במילון מתחילתו עד שמגיעים אל המלה המבוקשת;

2. <u>שיטת החיפוש</u> - בודקים מהי האות הראשונה של המלה, מסתכלים בתוכן העניינים באיזה עמוד מתחילות המלים עם האות המבוקשת, ניגשים אל החלק המתאים לאות זו במילון ושם מחפשים את המלה.

שיטה א׳ גוזלת זמן רב, כי בממוצע יש לעבור על מחצית המילון עבור כל מלה שמחפשים. בשיטה ב׳ ניתן למצוא את המלה המבוקשת ב-15 שניות בממוצע (וזה פחות בהרבה מאשר לקרוא את מחצית המילון...).

לו היו כותבים את המילון על סרט מגנטי מהיר, היו דרושות כ-45 שניות כדי לעבור על מחצית המילון ולמצוא את המלה המבוקשת, ואילו על דיסק מגנטי הגישה אל המלה המסוימת היא כמעט מיידית, בסדרי גודל של כחצי מאית של שנייה.

שיטת החיפוש השניה נקראת חיפוש לפי אינדקס, והיא יעילה ביותר לא רק בכל מה שנוגע לחיפוש מלים במילון, אלא גם בכל מה שנוגע לאיתור מוצר בכרטסת מוצרים, איתור אדם ברשימת שמות וכו׳.

קבצים המאוחסנים על דיסקים מגנטיים ניתנים לעיבוד במספר שיטות. כאן תתוארנה כמה מהן:

<u>9.5.1 שיטת ארגון אינדקס סדרתי</u>
INDEXED SEQUENTIAL ACCESS METHOD (ISAM)

בקובץ אינדקס סדרתי לכל רשומה חייב להיות מפתח (שדה מזהה מספרי, כגון מס׳ תעודת זהות, מספר חשבון וכו׳). הקובץ עצמו בנוי משלושה שטחים שונים: שטח עיקרי, ובו מאוחסנות הרשומות בצורה ממויינת, שטח האינדקסים בו רשומים מפתחות הרשומות ושטח גלישה.

בזמן <u>בניית הקובץ</u>, נכתבות הרשומות בשטח העיקרי בצורה ממוינת, החל מן הרשומה בעלת המספר הנמוך ביותר. בשטח אינדקס נרשמים מפתחות הרשומות ומקומם בדיסק. שטח הגלישה נשאר ריק.

בזמן <u>עיבוד הקובץ</u>, כאשר מתוספת רשומה חדשה, היא תוכנס למקומה המתאים בקובץ, בתוך כדי "הזזת הרשומות הצידה". רשומה, שאין

לה מקום במסילה, תוכנס לשטח הגלישה. האינדקס מציין אילו רשומות נמצאות בשטח הגלישה, כדי לאפשר עיבוד סדרתי.

ניתן לעבד קובץ של אינדקס סדרתי בשלוש דרכים:

• <u>עיבוד סדרתי</u>

עיבוד זה זהה לעיבוד קובץ סדרתי, כפי שתואר לעיל.

• <u>עיבוד אקראי</u>

בשיטת עיבוד זו פונים אל הרשומה הבודדת לפי המפתח שלה. התוכנה מאתרת את מקום הרשומה באינדקס, ואח"כ קוראת את הרשומה לתוך הזיכרון הראשי לשם עיבוד. לאחר העיבוד, נכתבת הרשומה בחזרה במקומה הקודם.

• <u>עיבוד אינדקס סדרתי</u>

בדרך עיבוד זו ניתן לשלב את שתי הדרכים הקודמות: כאן ניתן לגשת ישירות אל רשומה במקום כלשהו בקובץ, ולהתחיל לעבד את הרשומות משם והלאה בעיבוד סדרתי. בצורה זו נחסך זמן יקר של סריקת הקובץ. יתרון נוסף: אין צורך לעבד את הקובץ עד סופו, אלא ניתן להפסיק את העיבוד בכל מקום רצוי.

<u>9.5.2 שיטת ארגון לגישה ישירה</u>

DIRECT ACCESS METHOD (DAM)

בעוד ששיטת ISAM איפשרה הן עיבוד סדרתי והן עיבוד אקראי, הרי שיטת הגישה הישירה מתרכזת באחזור רשומות בשיטה האקראית. שיטת עיבוד זו באה כאן לידי ביטוי בצורה יעילה הרבה יותר.

ארגון הקובץ בשיטה זו מחייב יצירת קשר בין המפתח של הרשומה לבין מקומה הפיסי על הדיסק. פירוש הדבר, שבזמן <u>בניית</u> הקובץ דואגים לחשב את מקום הרשומה על הדיסק על פי נוסחה מסוימת או ליצור טבלה בה מופיעים המפתח כל רשומה והכתובת, בה היא נרשמה על הדיסק.

נקודות המאפיינות קובץ בגישה ישירה:

• הרשומות אינן מסודרות בקובץ לפי סדר הגיוני של המפתחות;

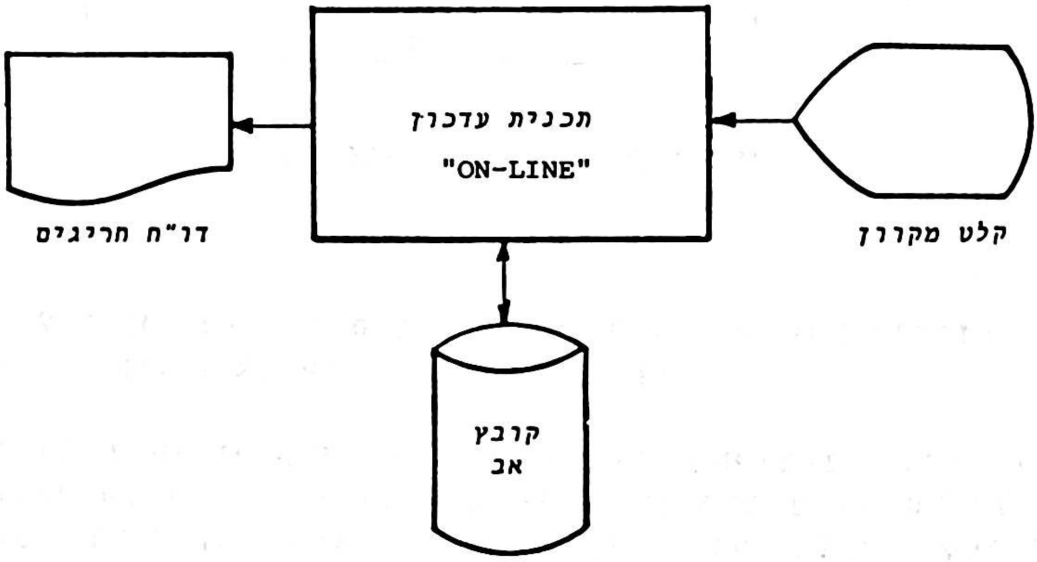
• מקומה הפיסי של כל רשומה על הדיסק נקבע על פי אלגוריתם מסוים, שנכתב על ידי התכניתן;

• בתוך כדי בניית הקובץ מתעוררות מספר בעיות, כגון: רשומות כפולות, מקום מסוים מוקצה לשתי רשומות שונות, מקום ריק

בין הרשומות וכו'. התכניתן חייב לדאוג לפתרון הבעיות האלה בתכנית הבנייה (טעינה) שלו.

• הבעיות, שצוינו בעניין בניית הקובץ יכולות להתעורר גם במועד מאוחר יותר, בזמן הוספת רשומה חדשה לקובץ.

קובץ מסוג זה מתאים במיוחד לשיטת הגישה הישירה ברשתות תקשורת. מסופים <u>מקוונים</u> (ON-LINE), אשר מחוברים אל המחשב כל הזמן, מאפשרים לעדכן כל רשומה באופן מיידי (ראה ציור 9-15).



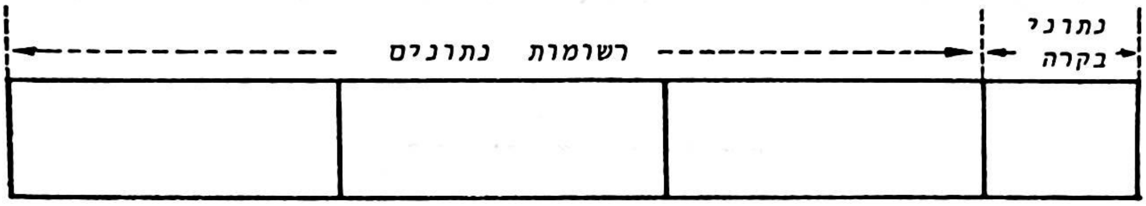
ציור 9-15 - עדכון קובץ DAM

יחידת הקלט יכולה להיות מכשיר בנקומט, יחידת מסך או מכונת כתיבה. תכנית העדכון מאפשרת דו-שיח בין מפעיל יחידת הקלט לבין המחשב. כל רשומה, אשר מגיעה מיחידת הקלט היא רשומת תנועה. לאחר בדיקות מתאימות על נכונות ועל תקיפות הנתונים (בדיקות לוגיות) תעדכן התכנית את רשומת האב, או תכתוב אותה כרשומה חריגה בקובץ מיוחד. רצוי לדאוג, לכך שמפעיל יחידת הקלט יקבל הודעה, המציינת מה עלה בגורל רשומת התנועה, אשר שלח אל המחשב.

<u>9.6 VSAM</u>

אחת מהשיטות המתקדמות לארגון נתונים במערכות יבמ היא VSAM (VIRTUAL STORAGE ACCESS METHOD). שיטה זו קיבלה את שמה הודות לעובדה, שתכניות העיבוד שלה פועלות בסביבה של זיכרון בפועל (VIRTUAL STORAGE) בלבד. שלוש תכונות חשובות מייחדות שיטה זו משיטות הארגון האחרות:

1. הקובץ ב-VSAM מאורגן על הדיסק המגנטי בשטחים בעלי אורך קבוע הנקראים **גושי פיקוח** (CONTROL INTERVAL). גודלם של גושי הפיקוח נקבע לפני בניית הקובץ, והוא אינו משתנה. לאחר מכן מאוחסנות הרשומות בגושי הפיקוח משמאל לימין. בקצהו הימני של הגוש נמצא קטע ובו **נתוני בקרה** המתייחסים לרשומות, אשר באותו גוש (ראה ציור 9-16).

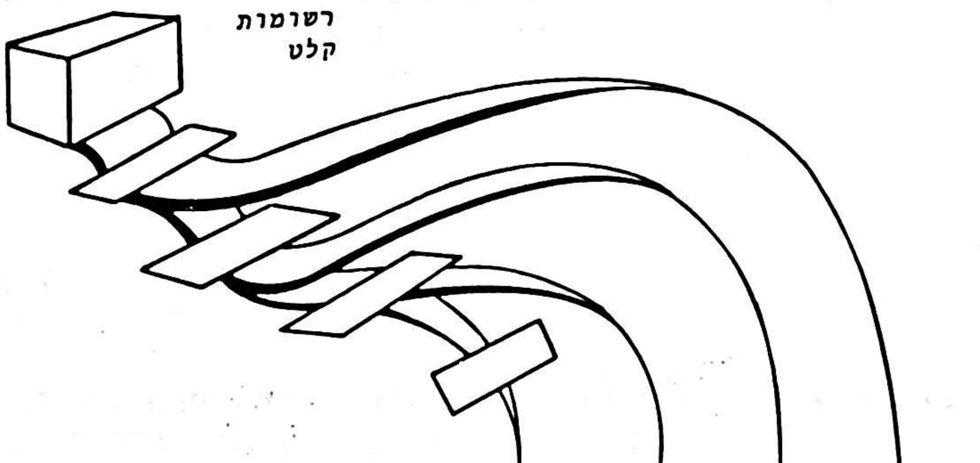


ציור 9-16 - גוש פיקוח (VSAM)

2. VSAM הינה השיטה היחידה התומכת בשלוש שיטות עיבוד: עיבוד סדרתי, עיבוד אקראי ועיבוד לפי אינדקס.

3. ניהול הנתונים נעשה ב-VSAM בעזרת קטלוגים. קיימים שני סוגים של קטלוגים: MASTER-CATALOG ובו מידע על כל קבצי VSAM במערכת, USER-CATALOG שבו מידע נוסף, לפי ראות עיניו של המשתמש. בכל מערכת חייב להיות MASTER-CATALOG אחד, ואילו מספרם של USER-CATALOGS אינו מוגבל.

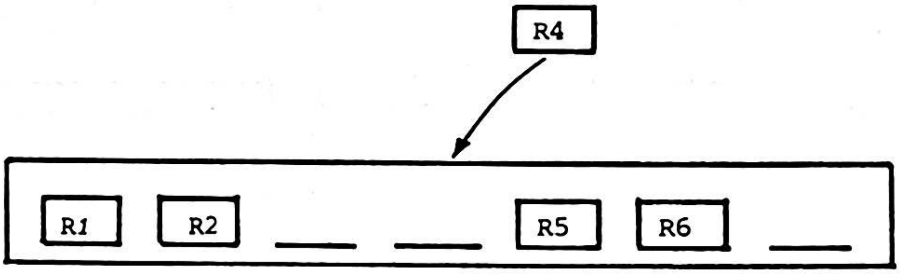
שיטת הגישה VSAM תומכת בשלושה סוגי קבצים, הנבדלים זה מזה בארגון הפנימי של הרשומות.



ציור 9-17 - קובץ ESDS

בקובץ מסוג ESDS (ENTRY-SEQUENCED DATA SET) נרשמות הרשומות לפי סדר הגעתן. רשומות חדשות מתוספות בסוף הקובץ הקיים (ראה ציור 9-17) ולכן בזמן כתיבת הרשומה אין צורך לציין מפתח או שדה זיהוי אחר. לאחר כתיבת רשומה חדשה מחזירה המערכת אל תכנית המשתמש את הכתובת היחסית של הרשומה (RELATIVE BYTE ADDRESS - RBA). המשתמש צריך לבנות טבלה או אינדקס מן הכתובות, ולהיות אחראי לטיפול ברשומות. שיטה זו מאפשרת עיבוד סדרתי של הקובץ.

קובץ RRDS (RELATIVE RECORD DATA SET) הינו אוסף של קטעים בעלי אורך שווה, המהווים מעין שרשרת של תאים ממוספרים, שגודלה נקבע בזמן יצירת הקובץ. כל רשומה המוכנסת לקובץ מקבלת מספר הנקרא מספר רשומה יחסי (RELATIVE RECORD NUMBER) ולפי מספר זה היא תאוחסן. למשל, רשומה 4 תיכנס לתא מס' 4, ואין זה חשוב, אם שאר התאים ריקים או מלאים. תכניות העיבוד בשיטת ה-RRDS מבצעות פעולות עדכון, הוספה או ביטול של רשומות לפי המספר היחסי של הרשומה. על כן ניתן לומר, שהעיבוד כאן נעשה לפי גישה ישירה (DIRECT ACCESS).



ציור 9-18 - קובץ RRDS

השיטה השלישית לארגון ולעיבוד קבצים ב-VSAM מבוססת על המפתח (KEY) בתוך הרשומה. הרשומות מסודרות בקובץ לפי סדר עולה של המפתח, ולכן סוג זה של קבצים נקרא KSDS (KEY-SEQUENCED DATA SET). VSAM בונה אינדקס של המפתחות (בדומה לשיטת ISAM) ולפיו אפשר לאתר רשומות בקלות ובמהירות.

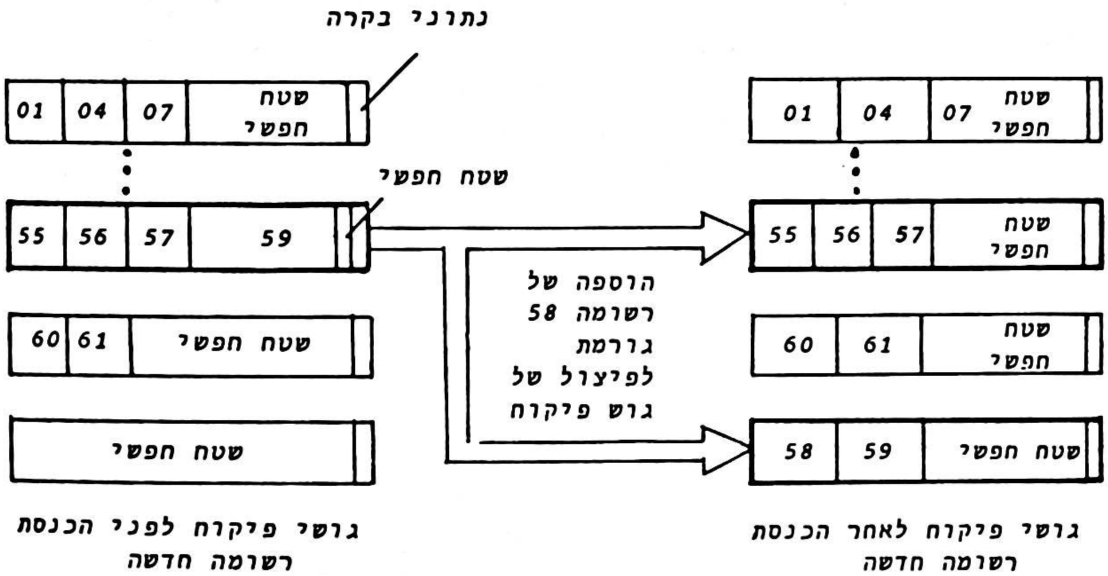
הוספת רשומות חדשות נעשית כאן בצורה דינמית, ע"י שימוש בשטח חפשי (FREE SPACE) ולא על-ידי שימוש בשטח גלישה, כפי שזה היה ב-ISAM. שטח חפשי הינו חלק מגוש הפיקות (ראה ציור 9-19). כל משתמש חייב לקבוע בזמן יצירת קובץ KSDS מהו אחוז השטח החפשי בכל גוש פיקות.

בקובץ KSDS מוכנסת רשומה חדשה לקובץ לפי המספר שבמפתח הרשומה. למשל, בציור 9-19 אם מתוספת רשומה בעלת מפתח 3, היא תיכנס בין רשומה 2 ורשומה 4, והשטח הריק יצטמצם.



ציור 19-9 - גוש פיקוח טיפוסי ב-KSDS

במקרה של מחסור בשטח חפשי, VSAM יבצע <u>חלוקה</u> (SPLIT) של גוש הפיקוח, ואל גוש הפיקוח החדש תועברנה הרשומות המתאימות (ראה ציור 20-9).



ציור 20-9 - פיזור שטח ריק

<u>יתרונות VSAM</u>

כפי שכבר צויין, אחד מהיתרונות של שיטת VSAM היא התמיכה בשלושה סוגי קבצים שונים, בהתאם לדרישות העיבוד השונות. יתרון נוסף של השיטה היא <u>התאימות</u> שלה. תאימות זו מתבטאת, בכך שקובצי VSAM ניתנים להעברה ממערכת הפעלה אחת לשנייה, ללא צורך בפעולת הסבה כלשהי. כן ניתן להעביר את הקבצים מסוג אחד של דיסקים לסוג אחר, מבלי שהמשתמש יצטרך לבצע שינויים בהגדרות הקובץ או בתכניות היישום.

יתרון בולט של שיטת ה-VSAM היא <u>הניצול היעיל</u> של משאבי המערכת. מכיון שהקובץ מאורגן בגושים בעלי אורך קבוע, אפשר לנצל את שטחי האחסון על הדיסקים ללא קשר עם סוג הדיסק. ההקצאה הדינמית של השטחים הריקים בגושי הפיקוח, אשר באה במקום הקצאת שטחי הגלישה ב-ISAM - מבליטה יתרון זה.

האחרונה, אך אולי החשובה בשורת היתרונות היא הגמישות בעיבוד הקבצים. כבר הוזכר ששיטת VSAM מאפשרת עיבוד סדרתי, עיבוד אינדקס-סדרתי (עולה או יורד!) ועיבוד אקראי, הכל על פי שדה מפתח ראשי או שדות מפתח משניים. אולם יש להדגיש עוד, שניתן לבצע שיטות עיבוד שונות בו-זמנית, באותו יישום עצמו. אפשר לבצע במקביל עיבוד סדרתי ועיבוד אקראי או שני עיבודים סדרתיים על חלקים שונים של אותו קובץ.

9.7 שיקולים לארגון הנתונים ועיבוד הקבצים

לפני הצגת השיקולים השונים לארגון ולעיבוד קבצים יש לציין כלל חשוב: אין טכניקה אחת ויחידה, אשר טובה לכל המצבים! בכל זאת, קיימים מספר שיקולים, אשר יכולים לסייע בהחלטה לטובת מבנה נתונים זה או אחר ובזכות שיטת עיבוד מסוימת.

1. שיקולים לגבי ארגון הנתונים

- סדר הרשומות בקובץ - בקבצים המשרתים יישומים מסוימים דרוש מיון של הרשומות. בקובץ הבנוי עם אינדקס הדבר הכרחי.

- אורך הרשומות יכול להיות קבוע או משתנה ביישומים שונים.

- מספר רשומות בגוש (BLOCKING FACTOR) - אם הרשומות נמצאות בגושים נודעת חשיבות רבה למספר הרשומות בגוש. זהו גורם בעל השפעה בעת העברת נתונים מיחידת הקלט/פלט לזיכרון הראשי ולהיפך. כמו כן יש לכך השפעה על ניצול יעיל של אמצעי האחסון החיצוני כגון, סרט או דיסק מגנטיים;

- מחיר אמצעי האחסון - סרט מגנטי זול יותר לאחסנה מאשר דיסק מגנטי;

- שיעור הגידול של הקובץ - גידול מהיר אינו מתאים למבנה ISAM בגלל שטחי הגלישה;

- שיעור דחיסת הנתונים - שיעור זה נמוך מאד במבנה DAM וגבוה ב-SAM;

2. שיקולים לגבי עיבוד הקבצים

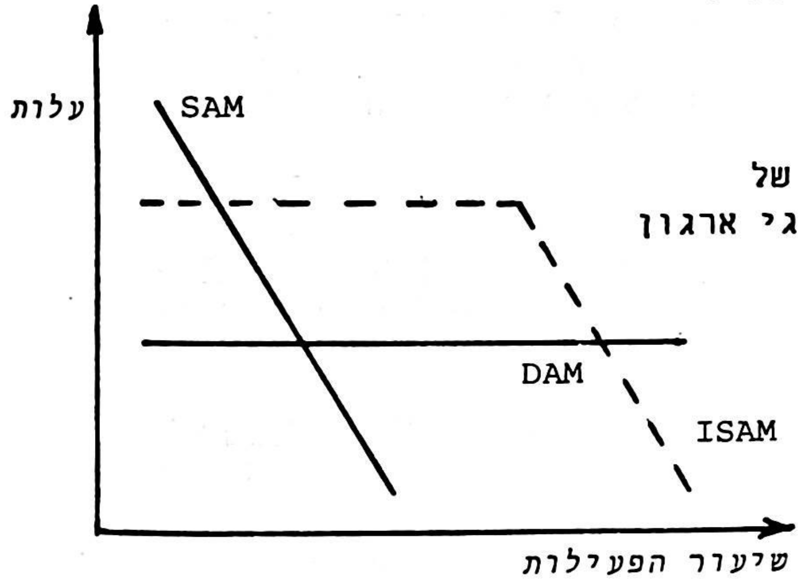
- מהירות גישה - זמן גישה ממוצע הינו קצר לגבי קבצים על דיסקים וארוך יותר לגבי סרטים מגנטיים. רק הדיסקים מאפשרים גישה ישירה, או לפי אינדקס;

• <u>דרישות עיבוד</u> - אם הדרישה היא לאפשרות גישה סדרתית וישירה - חייבים לעבוד ב-ISAM או בקבצי VSAM מסוימים.

• <u>ארגון מחדש של הקובץ</u> - נדרש ב-ISAM לעתים קרובות, ב-SAM פחות וב-DAM כמעט ולא.

• <u>שיעור הפעילות בקובץ</u> - קובץ אשר משמש לשאילתות אקראיות ואשר מחייב זמן תגובה מינימלי כמקובל בעיבוד מקוון, יהיה בנוי בשיטת גישה ישירה או באמצעות אינדקס.

הקשר בין שיעור הפעילות של הקובץ לבין ההוצאות הכרוכות בכך, מוצג בציור 9-21:



ציור 9-21 - השוואה של עלויות עיבוד לפי סוגי ארגון

טבלת הסיכום שבציור 9-22 מציגה את הקריטריונים להחלטה על שיטת הארגון ועל שיטת העיבוד של הנתונים.

<u>טבלת סיכום</u>

	ארגון סידרתי (SAM)	ארגון ISAM	ארגון ישיר DAM
עיבוד סידרתי וישיר	-	+	-
צורך ברה-ארגון כל פעם	-	+	-
שיעור דחיסה טוב (חסכוני באיחסון)	+	-	-
התאמה לגידול מהיר	+	-	++
התאמה לתקשורת	-	+	++
מהירות באתור רשומה	-	+	++
עלות לשליפת רשומה אקראית	++	+	-

ציור 9-22 - טבלת סיכום לשיטות ארגון ועיבוד נתונים

פרק 9 - שאלות

1. כאשר אומרים "ניהול נתונים" - למה מתכוונים בדיוק?

2. מה מאפיין ארגון נתונים סדרתי?

3. הסבר את המונח האנגלי BLOCKING FACTOR.

4. הבחן בין יצירת הקובץ לבין עדכונו.

5. האם קובץ התנועות חייב תמיד להיות ממויין לפני פעולת העדכון? הסבר.

6. באיזה מקרה היית משתמש בקובץ סדרתי?

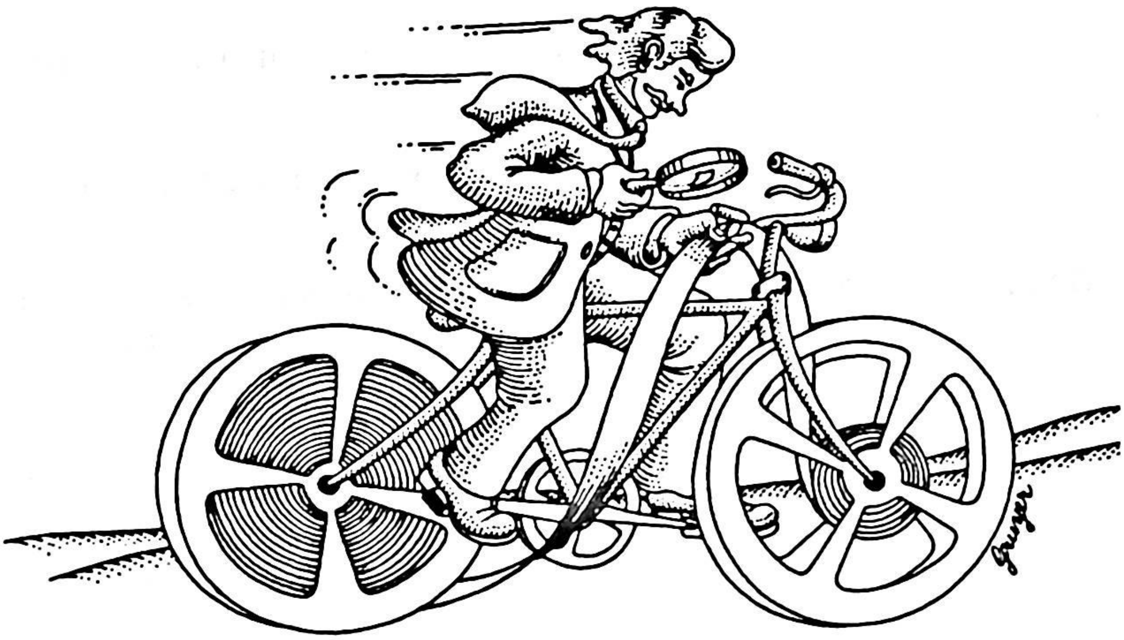
7. ציין את יתרונות השימוש בגישה ישירה.

8. האם אפשר לוותר, לדעתך, על רשומות באורך משתנה? נמק.

9. טוענים, ששיטת VSAM בזבזנית. התייחס לטענה זו.

10. מהו תפקיד שדה המפתח (KEY) בקבצים? האם כל שדה יכול להיות שדה מפתח?

11. לבעל חנות תקליטים גדולה יש מחסן גדול, ובו כ-10,000 תקליטים שונים. בעל החנות מעונין לדעת בסוף כל שבוע, אילו תקליטים נמצאים במחסן לקראת המכירות בשבוע הבא. אל המחשב של החנות ניתן לחבר גם סרטים מגנטיים וגם דיסקטים. הגש הצעה מנומקת באיזה סוג של ציוד עליו להשתמש.



פרק 10 - בסיסי נתונים

בארצות רבות נמנעים, מסיבות של צנעת הפרט, מהקמה של מאגר נתונים מרכזי, אשר מכיל את כל הנתונים על האזרח. אולם במדינות אחדות קיימים מאגרי נתונים כאלה, ולפיכך ייתכן, שבמשרד הפנים של מדינה כלשהי נאגר מידע רב על כל תושבי המדינה. כל המידע נמצא בקובץ גדול מאד, בו כל רשומה מכילה את הפרטים המזהים של כל תושב, כפי שזה מוצג בציור 1-10. בנוסף לפרטים המזהים הללו, מופיעים שדות נוספים ברשומה, ובהם מידע על זכות בחירה, מספר הקלפי, רקע חינוכי, תעסוקה ועוד.

מס. ת.ז.	מען	שנת לידה	מקום לידה	מין	מצב משפחתי

ציור 1-10 - דוגמת רשומה במשרד הפנים

חשוב לציין, שגם במשרדי ממשלה אחרים, מלבד משרד הפנים, קיימים קבצים, ובהם פרטים רבים על תושבי המדינה. במשרד האוצר נאגר מידע על מצבו הכלכלי של התושב (רכוש, משכורת, מסים, חסכונות, מלוות וכוי), במשרד הביטחון - פרטים על השירות הצבאי, במשרד החינוך - על שלבי ההשכלה השונים, במשרד הרישוי - פרטים על רשיון נהיגה ובעלות על רכב (אם יש) וכוי.

בדיקה פשוטה תראה לקורא, שהפרטים על כל תושב רשומים במשרדי ממשלה אחדים, כאשר בחלק הראשון של הרשומה מופיעים הפרטים המזהים, ובחלק השני - המידע המיוחד, הנוגע לפעילות אותו משרד. מצב זה מוצג בציור 2-10 :

משרד הפנים

מס' ת.ז.	שם	מען	שנת לידה	מקום לידה	מין	מצב משפחתי	מס' דרכון	מס' קלפי
----	--	---	----	----	---	------	------	-----

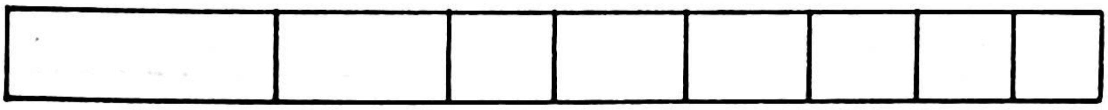
פרטים מזהים ← → פרטים הנוגעים למשרד הפנים

משרד האוצר

פרטים מזהים + פרטים הנוגעים למשרד האוצר.

משרד הבטחון

פרטים מזהים + פרטים הנוגעים למשרד הבטחון.



ציור 2-10 - רשומות במשרדי ממשלה שונים

כל קובץ תוכנן ואורגן במיוחד ליישום מסוים לפי הצרכים של אותו משרד, אשר הקים אותו. כל משרד דאג להחזקת הקובץ שלו בעצמו, בשיטות ובמועדים, שנראו לו. מצב כזה קיים בארצות רבות בעולם, ואין ישראל יוצאת דופן.

עתה נתאר לעצמנו, שאזרח משנה את מקום מגוריו. כאזרח שומר חוק, הוא מודיע למשרד הפנים באמצעות טופס רשמי על כתובתו החדשה. טופס כזה אפשר למלא בכל משרד אזורי של משרד הפנים, ושם רושמים בתעודת הזהות את המען החדש. נשאלת השאלה, האם הכתובת החדשה מועברת אל משרדי הממשלה האחרים? התשובה שלילית. להיפך, האזרח מתבקש להודיע <u>בנפרד</u> למשרד הרישוי (שאם לא כן, לא יחדשו לו את רשיון הנהיגה), למשרד הביטחון או ליחידה הצבאית (ולא, הוא צפוי למשפט). היה רצוי שמשרד הפנים יודיע <u>לכל</u> שמונת או תשעת משרדי הממשלה, בהם הפרט הזה קיים בקובץ, אולם המציאות היא שונה. משרד הפנים, שקיבל את ההודעה, מעדכן את הקובץ שלו בלבד. בקבצים האחרים מופיעה עדיין הכתובת הישנה.

10.1 ריבוי קבצים לעומת בסיס נתונים

ניתוח של המצב שתואר קודם, מצביע על חסרונות בשיטה הקיימת של ריבוי קבצים:

1. <u>כפילות נתונים</u>

בכל משרדי הממשלה רשומים אותם פרטי הזיהוי, שמונה או עשר פעמים ואולי יותר.

2. <u>עדכון חלקי של הנתונים</u>

חלק מן הנתונים מתעדכן מדי פעם, כמו למשל הכתובת. אולם עדכון זה חלקי, כי הוא מבוצע רק במשרד אחד או שניים מבין משרדי הממשלה.

3. <u>נתונים מיותרים</u>

ייתכן שבקבצים השונים, <u>בחלק המיוחד</u> של כל רשומה, ישנם שדות, שגם הם מופיעים בכמה משרדי ממשלה.

4. <u>תחזוקת הנתונים</u>

קשה לתחזק את הנתונים, המפוזרים במשרדים רבים. קשה לדאוג, שהנתונים יהיו מעודכנים בכל הקבצים בעת ובעונה אחת.

5. <u>קושי בקבלת תמונה כוללת</u>

אם רוצים לקבל מידע אשר נוגע לכלל האוכלוסיה, כגון: מהו שיעור הגברים והנשים בין הגילים שמונה-עשרה לשלושים, שהם רווקים, ומשכורתם מתחת לרמה מינימלית מסוימת, הדבר קשה ומסובך. המידע מפוזר בקבצים רבים, המפוזרים במקומות שונים, רחוקים זה מזה. הגישה אל הקבצים הללו נפרדת, דרך מערכות נפרדות.

מתיאור החסרונות של המצב במשרדי ממשלה, מתברר שרצוי שינוי משמעותי ומעמיק בכל הנוגע לאחזקת הנתונים ולעיבוד שלהם. אחד מהפתרונות אשר נראה מתאים, הינו <u>איחוד</u> הקבצים הקיימים לקובץ אחד גדול, בו הרשומה תיראה כך:

פרטי הזיהוי של התושב	פרטים הנוגעים למשרד הפנים	פרטים הנוגעים למשרד האוצר	פרטים הנוגעים למשרד הבטחון

ציור 3-10 - רשומת תושב חדשה

הקובץ החדש הזה יכול לשמש <u>בנק של נתונים</u> (DATA BANK) או <u>בסיס נתונים</u> (DATA BASE). אפשרי למנות משרד ממשלתי אחד ויחיד, אשר יהיה אחראי על הקמת בסיס הנתונים, עדכונו ואחזקתו. שיטה חדשה זו יכולה לענות על כל הצרכים ויחד עם זאת לבטל את החסרונות, שהוזכרו לעיל:

* <u>מניעת הכפילות בנתונים</u> - כל פרט יופיע פעם אחת בלבד, במקום שמונה או עשר פעמים. בשיטה כזו גם יחסך מקום באמצעי אחסון (סרטים או דיסקים מגנטיים).

* <u>עדכון מרכזי של הנתונים</u> - רק גורם מרכזי אחד, אשר אחראי על הקובץ מורשה לעדכן את הנתונים. הנתונים המעודכנים יעמדו לרשות <u>כל</u> המשתמשים. כך יוצא, שאם אזרח יודיע על שינוי כתובת, כתובת חדשה זו תהיה מוכרת <u>לכל</u> משרדי הממשלה ומייד, עם העדכון.

* <u>אחזקת הנתונים</u> - גורם מרכזי אחד מורשה לטפל בנתונים, פירוש הדבר יעול בהקמת בסיס הנתונים, וטיפול מהיר ומהימן בכל הנוגע להוספה, ביטול או עדכון התוכן של הרשומות.

* <u>קבלת תמונה כוללת</u> - אפשרות לקבל מידע כולל או חתך של האוכלוסיה, לפי הצורך.

<u>הגדרה</u>

<u>בסיס נתונים (DATA BASE)</u> הינו אוסף נתונים אשר מאורגנים במבנה מיוחד, המאפשר קשרים פנימיים בין הנתונים, ואשר משמש מקור אחד ויחיד ליישומים רבים.

או

<u>בסיס נתונים (DATA BASE)</u> הינו אוסף נתונים מאורגנים, שיש קשר לוגי ביניהם, ואשר ניתנים לעיבוד על ידי יישום אחד או יותר.

<u>10.2 תכנון בסיסי נתונים</u>

בגישה המסורתית לניהול נתונים היתה האחריות לנתונים בידי המחלקות, או היישומים השונים. וכפי שראינו, לכל יישום או לכל מסגרת ארגונית היה קובץ נתונים אשר התאים לצרכיו. בעית השליטה על קבצי הנתונים השונים חייבה פתרון בצורה של ריכוז, שליטה ובקרה, והעיקר - הכרה בנתונים כמשאב של הארגון. הכרה זו הביאה להקמה של בסיס הנתונים ויצרה תפקיד חדש בארגון, התפקיד של <u>מינהלן בסיס הנתונים</u> (DATA BASE ADMINISTRATOR - DBA).

מינהלן בסיס הנתונים, או המינהלנים של בסיסי הנתונים השונים, <u>אינם</u> הבעלים של הנתונים. הם אחראים לניהולם התקין והיעיל ולאבטחתם. תוכן הנתונים במאגר נשאר באחריות היישומים השונים, אשר כוללת הזנה, עדכון והרשאת שימוש בהם. מינהלן בסיס הנתונים מנחה בדרך כלל את הצוות אשר מקים את בסיס הנתונים ומתחזק אותו.

הקמה או בנייה של בסיסי נתונים הינה אחד מהתפקידים החשובים, המורכבים והמתמשכים, שמוטל על צוות עיבוד הנתונים. תכנון נכון של בסיס הנתונים הינו תנאי הכרחי להצלחת הפרויקט של ההקמה. ארגון בסיס הנתונים, תכנון השדות, וקביעה אילו נתונים ייכללו בכל רשומה חייבים להיעשות בתאום מלא ובשיתוף פעולה הדוק עם כל הנזקקים לשרותים של בסיס הנתונים. נמחיש זאת בדוגמה קצרה.

נניח שרוצים להקים בסיס נתונים, אשר בין היתר, ישמש את היישום של ניהול המלאי של המחסן. דוגמה מכרטסת המלאי של מחסן מוצגת בציור 10-4.

מס' קטלוגי	שם פריט	כמות במחסן	כמות בהזמנה	מחיר ליחידה	מיקום
45678	פטיש 1/2 ק"ג	20	50	34	B8
58210	מברג "2	12	20	15.5	C2

ציור 10-4 - רשומת מלאי

המחסנאי ישמח, אם במקום הכרטסת יקבל מסוף, המחובר למחשב, שבאמצעותו יוכל לבצע שאילתות, ואפילו עדכונים. אולם האם המחסנאי יודע מתי להזמין כמות נוספת מפריט מסוים? האם הוא יודע, בשלב זה, מהו הערך הכולל של פריט מסוים, או קבוצת פריטים מסוימת? האם הוא יודע מהי רמת המינימום של כל פריט (המחייבת הזמנה אוטומטית), והאם רמה זו נכונה? כיצד היא נקבעה? יש להניח, שחלק מן השאלות הללו יישאר ללא מענה.

צוות ההקמה של בסיס הנתונים אינו יכול לקבוע את התשובות באופן שרירותי. דרושה כאן חקירה יסודית של המצב, לימוד הנושא בתוך כדי שיתוף פעולה הדוק עם המשתמש. יש לבדוק, מהם הנתונים הנוספים, שחייבים להיכלל בבסיס הנתונים, ומי יקבע את היקפם וטיב הקשר שלהם עם הנתונים האחרים של הארגון. התכנון הסופי של בסיס הנתונים ייעשה רק לאחר קבלת תשובות מספקות ומספיקות לכל השאלות הללו.

10.3 הקמת בסיסי נתונים

הקמה, או יצירה של בסיס נתונים תיעשה בעזרת תוכנה מיוחדת. קיימים סוגים רבים ושונים של מערכות תכנה, המיועדות לבצע משימה זו, ששמן הכולל הוא מערכות לניהול בסיסי נתונים - DBMS (DATA BASE MANAGEMENT SYSTEMS). תיאור המערכות הללו חורג

ממסגרת הספר הזה. נזכיר שמות של אחדות מהן, הנפוצות גם בישראל: SQL/DS, IDMS, DL/1, DB/1, TOTAL, ADABAS, IMS. מערכות אלו נועדו ליצור מסגרת, אשר מצד אחד הינה מסגרת נוחה לאחזקת הנתונים ולטיפול בהם, ואשר מצד אחר מאפשרת דרכי גישה שונות אליהם, בהתאם לצרכים המגוונים של המשתמשים הרבים. מטרה זו מושגת ברוב המערכות, בעזרת כמה "טכניקות", אותן נתאר בקצרה:

1. <u>חלוקת הרשומה לסגמנטים</u>

<u>סגמנט</u> (SEGMENT) הינו היחידה הבסיסית של בסיס הנתונים. הוא יכול להכיל שדה אחד או מספר שדות, הקשורים ביניהם קשר לוגי מסוים (ראה ציור 5-10). הסגמנטים יכולים להיות שונים באורכם, ומספרם ברשומה אינו קבוע.

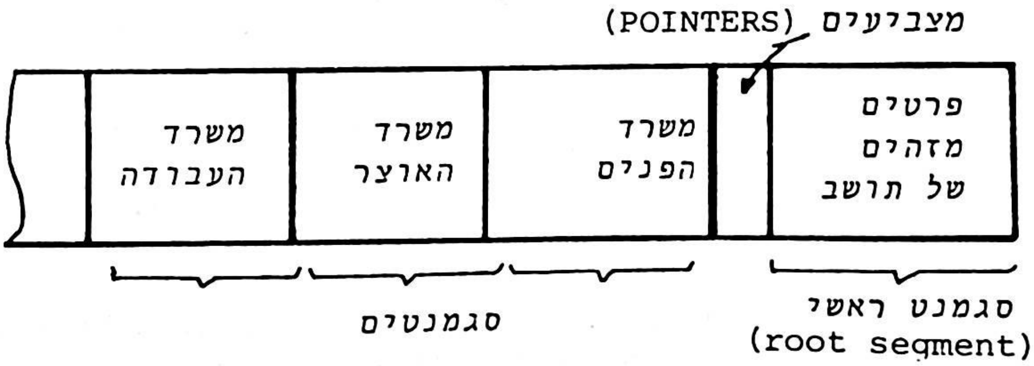
2. <u>רגישות (SENSITIVITY)</u>

רגישות הינה אמצעי, שבעזרתו ניתן להגדיר באיזה סגמנט של רשומה תורשה תכנית מסוימת להשתמש, ואילו פעולות ניתן לבצע עם סגמנט מסוים (עדכון, הוספה, ביטול וכו'). אמצעי זה מאפשר למשתמש קצה (בעזרת תכנית יישום מתאימה) לראות רק סגמנטים מסוימים, ולא לדעת כלל על קיום סגמנטים אחרים ברשומה. לדוגמה, ניתן לכתוב תכנית, אשר מאפשרת לעובדי מחלקת המנגנון לשלוף רק את הסגמנטים המזהים של רשומת העובד (מס' תעודת זהות, שם העובד) וסגמנט הכתובת. יישום אחר של מדור השכר, למשל, יאפשר לשלוף ולעבד את הסגמנטים המזהים וכל הסגמנטים הקשורים בשכר. סגמנטים אחרים לא יוצגו והם יהיו <u>"שקופים"</u> (TRANSPARENT) למשתמש שאינו מורשה "לגשת" אליהם.

<u>10.4 מבנים לוגיים של בסיסי נתונים</u>

קיימים סוגים שונים של בסיסי הנתונים, הנבדלים זה מזה במבנה הלוגי שלהם. נזכיר את המבנים הנפוצים ביותר.

1. <u>מבנה שטוח - FLAT (OR COMBINED) DATA MODEL</u>



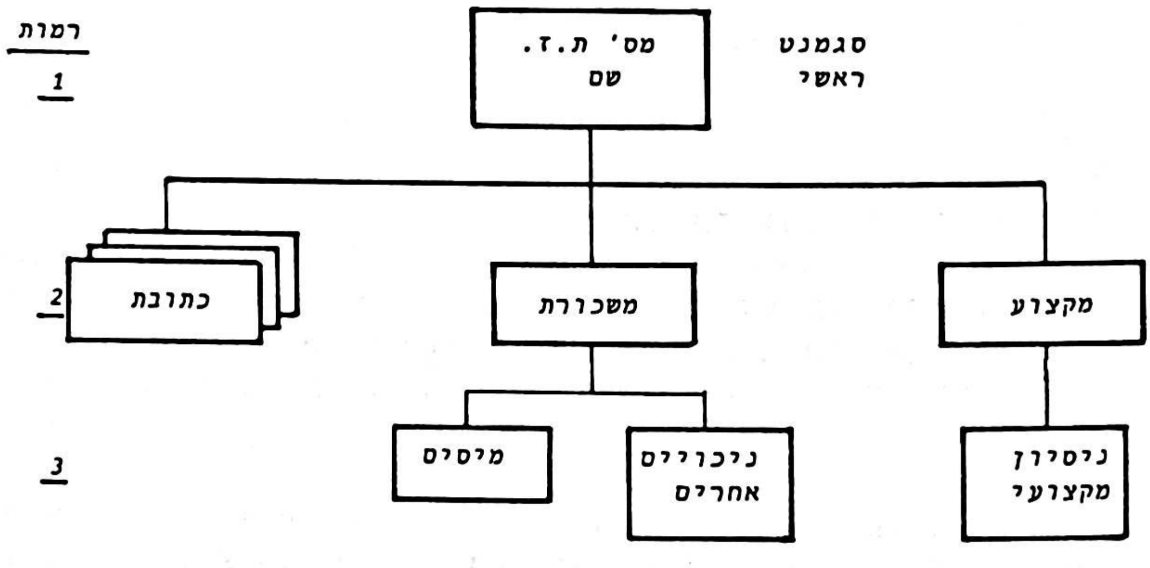
ציור 5-10 - רשומה בבסיס נתונים שטוח

בציור 10-5 מוצגת רשומת תושב של מבנה נתונים "שטוח". הרשומה מחולקת לסגמנטים. הסגמנט הראשון נקרא סגמנט ראשי, והוא מכיל את הפרטים המזהים כמו: מס' ת.ז., שם, כתובת וכו'. כל סגמנט נוסף מכיל מספר שדות, שיש ביניהן קשר לוגי מסוים. ברשומה זו, למשל, סגמנט של משרד האוצר יכיל את כל השדות, שהם בעלי תוכן רלוונטי לפעילות משרד האוצר.

בין הסגמנט הראשי והסגמנטים האחרים נמצאים <u>המצביעים</u> (POINTERS). לכל סגמנט יש מצביע, המאפשר לשלוף אותו על ידי תכנית השליפה. המצביע הוא למעשה הכלי, שבאמצעותו בא לידי ביטוי עקרון הרגישות (או "השקיפות") של הסגמנטים. במבנה השטוח כל סגמנט איננו תלוי בסגמנטים אחרים שברשומה, פרט לתלות בסגמנט הראשי, שבעזרתו נשלפים הסגמנטים.

2. <u>מבנה היררכי או מבנה עץ - HIERARHICAL DATA BASE MODEL</u>

הנתונים במבנה ההיררכי מסודרים בסגמנטים, אשר תלויים זה בזה ומסודרים באופן לוגי, <u>ברמות</u> (LEVELS) שונות. בציור 10-6 מוצג מבנה כזה, הקרוי גם <u>מבנה-עץ</u> (TREE STRUCTURE).



ציור 10-6 מבנה-עץ

כל מלבן, המופיע בציור 10-6 מייצג סגמנט, אשר יכול להכיל מספר שדות. סגמנט הכתובת, למשל, יכיל: שם ישוב, שם רחוב, מסי בית, מיקוד. אורך כל סגמנט יכול להיות שונה מזה של סגמנט אחר.

<u>הקשר הלוגי</u> בין הסגמנטים מוצג בעזרת הקווים המופיעים ביניהם. כמו כן ניתן להבחין <u>ברמות</u>. סגמנט ברמה מסוימת נקרא <u>אב</u> או <u>הורה</u> (PARENT) של סגמנט ברמה נמוכה ממנו, אם קיים קשר לוגי (קו מחבר) ביניהם (למשל סגמנט "משכורת"

הינו הורה של סגמנט "מיסים" בציור 10-6). ולהיפך: סגמנט מרמה נמוכה הוא סגמנט <u>בן</u> או <u>ילד</u> (CHILD) של סגמנט מרמה גבוהה יותר, אשר קשור אליו. כל הסגמנטים יחד הם רשומה של בסיס הנתונים.

יש להזכיר עוד את המושג <u>מופע</u> (OCCURANCE) של סגמנט. ייתכן, שסגמנט מסוים מופיע מספר פעמים, כמו למשל כל הכתובות של העובד בעשר השנים האחרונות, או כל ההשתלמויות המקצועיות שעבר במפעל.

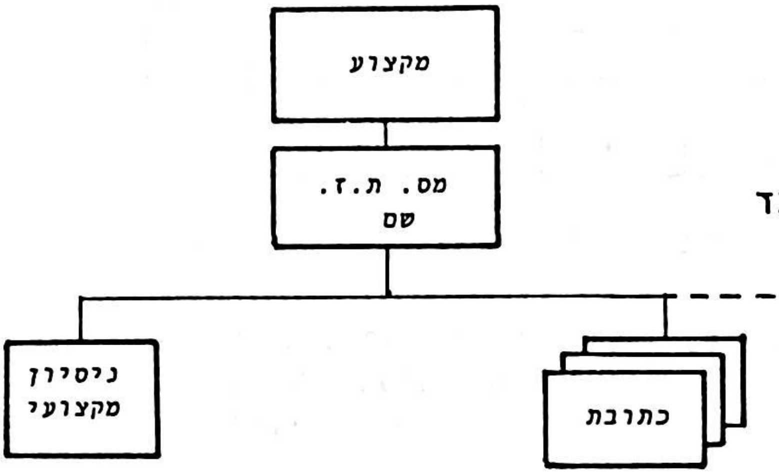
נשאלת השאלה, כיצד מסודרת רשומה של בסיס נתונים על הדיסק המגנטי. נושא זה נמצא בשליטה של <u>מערכת התכנה לניהול בסיסי נתונים</u> (DBMS - DATA BASE MANAGEMENT SYSTEM) ולא נדון בכך כאן. הצגה גרפית של <u>המבנה הפיסי</u> של רשומת בסיס הנתונים תתרום, כך אנו מקווים, להבנת הנושא. בציור 10-7 מוצגת הרשומה מציור 10-6 אך הפעם מנקודת ראות פיסית, זאת אומרת - כפי שהסגמנטים מסודרים על הדיסק המגנטי.

מס. ת.ז.	שם	כתובת 1	כתובת 2	כתובת 3	משכורת	מיסים	ניכויים אחרים	מקצוע

ציור 10-7 - הרשומה הפיסית

בציור 10-7 לא הוצגו <u>המצביעים</u> המאפשרים שליפה סלקטיבית של הסגמנטים השונים מתוך הרשומה. כזכור, סלקטיביות זו קשורה קשר הדוק בהגדרת <u>הרגישות</u> (SENSITIVITY) של הסגמנטים השונים. בעזרת הכלים הללו לא זו בלבד, שניתן <u>להפריד את המבנה הפיסי</u> של הרשומה על הדיסק מן המבנה הלוגי, אותו רואה המשתמש, אלא ניתן ליצור מבנים לוגיים נוספים.

ועדת ההשתלמויות של המפעל תוכל לראות את רשומת העובדים בצורה שונה מאשר מחלקת כח-אדם, כפי שמוצג בציור 10-8. ראייה זו מתאפשרת, כמובן רק לאחר שחבר ועדת ההשתלמויות (אחד המשתמשים, שרואיינו לפני הקמת בסיס הנתונים) הציג את דרישותיו, וצוות ההקמה, אשר כולל את <u>מנהלן בסיס הנתונים</u> (DBA) הגדיר מבנה לוגי נוסף בעת יצירת בסיס הנתונים. המלבנים המקווקוים מציינים את שאר הסגמנטים ברשומה, שהם "שקופים" במקרה זה לאנשי הועדה הזו.

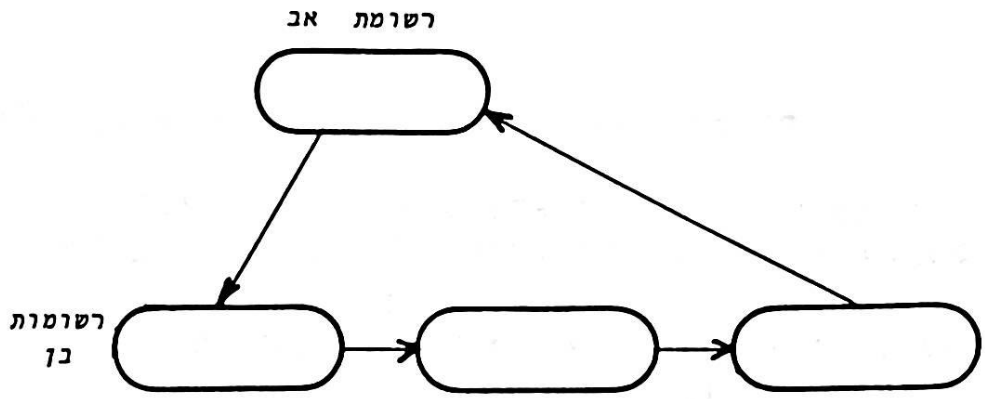


ציור 10-8 - רשומת עובד

3. <u>מבנה רשת</u> - <u>NETWORK DATA MODEL</u>

מבנה הרשת הוגדרה בצורה פורמלית בתחילת שנות ה-70, על ידי צוות עבודה בינלאומי בנושא בסיסי נתונים.

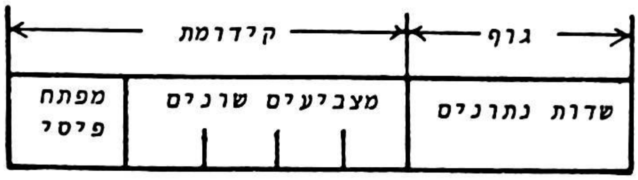
המודל הזה מאופיין על ידי שני סוגי רשומות: <u>רשומת אב</u> (OWNER RECORD) <u>ורשומת בן</u> (MEMBER RECORD). הקשר בין שני סוגי הרשומות הוא קשר של "אחד לרבים". כלומר, לכל רשומת אב עשויות להיות רשומת בן אחת או יותר. לדוגמה, "עובדים במחלקה" יצויינו ע"י רשומת המחלקה, שתופיע פעם אחת, ורשומות העובדים, אשר תופענה פעם אחת או יותר לפי מספר העובדים במחלקה. רשומת אב יכולה להיות קשורה למספר סוגי רשומות בן, ורשומת בן יכולה להיות קשורה לרשומות אב שונות.



ציור 10-9 - מבנה רשת

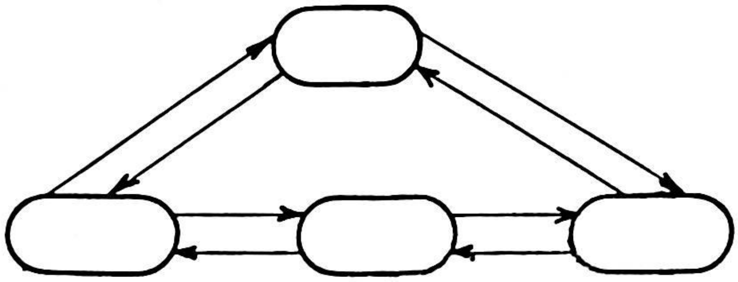
במודל הרשת מקובל המושג <u>חבורה</u> (SET) כדי להגדיר קשר בין שני סוגי רשומה, או יותר. אין הכוונה כאן להגדרה המתמטית של חבורה, אלא להגדרה של הצירוף של רשומת אב אחת ורשומת בן אחת או יותר.

הקשר בין הרשומות נעשה באמצעות מצביעים (ראה ציור 10-10). המצביעים יוצרים מבנים שונים של חבורות: מבנה של <u>שרשרת</u> (CHAIN), <u>טבעת</u> (RING), או <u>רשומת מצביעים</u> (POINTER ARRAY).



ציור 10-10 - מבנה של רשומה עם מצביעים

המצביעים בחבורה במבנה של שרשרת או טבעת יכולים להיות דו-כיווניים, ובכך הם מאפשרים סריקה של הרשומות בחבורה בשני הכיוונים. רשומת המצביעים מאפשרת גישה ישירה אל כל רשומה.



ציור 10-11 - חבורה עם מצביעים דו-כיווניים

כפי שנוכל לראות, החבורה הינה מבנה הירכי פשוט, בעל שתי רמות בלבד. בעזרת מבנה זה אפשר לתאר מבני נתונים מורכבים ביותר, אשר לא נציגם כאן.

4. מבנה טבלאי - RELATIONAL DATA BASE MODEL

מבנה נתונים טבלאי הוצג לראשונה בשנת 1970 על ידי ד"ר אדוארד קוד (E.CODD), אולם החל להיות נפוץ רק בתחילת שנות ה-80. כזכור, בסוף שנות ה-70 גדל השימוש במסופים במידה ניכרת, והפנייה אל המחשב דרך מסוף הפכה להיות נחלת הכלל. המסוף אינו עוד כלי עבודה בלבדי של "נבחרי" היחידה לעיבוד הנתונים. אחת מהתכונות של מבנה נתונים זה הינה "ידידותו" למשתמש, אשר מאפשרת לו שליפת נתונים על ידי שימוש בארבע הוראות פשוטות בלבד.

ד"ר קוד קבע מבנה, שבו מוצגים הנתונים למשתמש כאוסף של טבלאות. בכל טבלה השורות האופקיות הן הרשומות, ואילו העמודות הן השדות של כל רשומה.

בציור 10-12 מוצג קטע מבסיס נתונים טבלאי בשם STOCK.

מספר פריט PART-NO	שם פריט DESCRIPT	מחיר PRICE	כמות QUANT	שם ספק SUPPLIER
12234 12844 13307	חולצת גבר שמלה חלוק בית	400 980 970	85 120 61	אתא כיתן כיתן

ציור 10-12 - בסיס נתונים טבלאי

החשיבות של המבנה הטבלאי היא **בפשטותו**. פשטות זו מתבטאת לא רק בהצגת הסגמנטים, או הרשומות, אלא גם באפשרויות העיבוד על ידי המשתמש. הפנייה לקבלת מידע סלקטיבי לפי רצון המשתמש קלה ביותר. נניח שרוצים לקבל את כל הפריטים ושמות הספקים, בהם הכמות קטנה מ-100. נכתוב זאת כך, בשפת SQL/DS של יבמ:

```
SELECT NAME, SUPPLIER
FROM STOCK
WHERE QUANTITY < 100
```

כתוצאה, מקבלים טבלה חדשה:

שם פריט	ספק
חולצת גבר	אתא
חלוק בית	כיתן
חצאית	אתא

קיימים אופרטורים נוספים, כמובן, המאפשרים מיון טבלה קיימת לפי עמודות מסוימות, או מיזוג בין טבלאות. בכל מקרה התוצאה היא טבלה חדשה, שמוצגת על המסך בתשובה לשאילתה. הפשטות של השפה מאפשרת לכל משתמש לנסח בעצמו את השאלות, וליצור דו"חות או מסכים כרצונו.

10.5 היתרונות של בסיסי נתונים

בסיסי הנתונים הפכו להיות אחד מן המשאבים החשובים של כל ארגון שברשותו מערכת עיבוד נתונים ממוכנת. היות שהגישה אל בסיס הנתונים הפכה להיות נחלת רבים, מן הראוי לסכם את היתרונות בשיטת עבודה זו:

1. מניעת כפילות נתונים.

2. מניעת אי-התאמה של נתונים.

3. הנתונים בהישג ידו של כל דורש, באמצעות שיטות גישה מתקדמות.

4. אי-תלות בנתונים. המשתמש איננו "עבד" של מערכת המחשב, ואין לו צורך ללמוד את המבנה הפיסי של הנתונים. הוא מטפל רק במבנה הלוגי שלהם.

5. ריכוז נתונים המאפשר:

   - עיבוד סטנדרטי
   - גישה מבוקרת
   - הגנה על סודיות הנתונים

6. מניעת "סתירות" בנתונים (לא ייתכן מצב, בו לתושב מסוים תהיינה שתים עד שלוש כתובות בו-זמנית, ואולי שונות).

7. הגנה על שלימות הנתונים, בעזרת תוכנה מיוחדת.

8. הבטחת שלימות הנתונים ואמינותם.

9. אפשרות נוחה להרחבת בסיס הנתונים בלי להפריע לעבודה השוטפת.

10.6 בעיות אפשריות

למרות היתרונות, שנמנו בסעיף הקודם, אין להתעלם ממספר בעיות, הקשורות בשימוש בבסיסי נתונים. להלן תיאור כמה מהן:

1. <u>העברת נתונים בלתי מבוקרת</u> מבסיס נתונים אחד לאחר: כיום יש בסיסי נתונים בארגונים שונים, שבהם מופיעה רשומת תושב במבנה כמעט זהה (קופת-חולים, בנקים, משרדי ממשלה, המוסד לביטוח לאומי ועוד). כל מוסד כזה דואג לעדכון בסיס הנתונים שלו בלבד. אם תהיה העברת מידע ממוסד אחד לאחר, והעברה זו לא תהיה ממוסדת וכפופה לנוהלים ברורים וקבועים, ייתכן מצב, בו העברה בלתי מבוקרת של נתונים תגרום שוב ל"סתירות" במידע.

2. <u>אפשרות זיופים</u>. אם יהיה מעבר בין בסיסי נתונים, ובהנחה שכל בסיס נתונים מתעדכן ע"י גורמים שונים, תהיה אפשרות קלה יחסית לחדירה לתוך המערכת ולזיוף נתונים מסוימים.

3. פגיעה בצנעת הפרט. אם יאוחדו כל קובצי הממשלה לבסיס נתונים מרכזי אחד, יש חשש שקל יהיה יותר עכשיו, מאשר קודם, לראות את כל המידע, שהצטבר במשך שנים על אדם מסוים.

צנעת הפרט הינה הנושא הרגיש מבין השלושה, לאחריו נושא הזיופים והאחרון הוא נושא העברת הנתונים. נתאר לעצמנו מקרה דמיוני כזה: אזרח פלוני אלמוני היה מורה בבית-ספר תיכון בדרום הארץ. הוא נתן גם שיעורים פרטיים, אך לא הצהיר עליהם במועד לשלטונות מס ההכנסה. לאחר הוודע המקרה, הטיל עליו פקיד השומה קנס כספי בצירוף תשלומי מס מיוחדים בגין אותם שיעורים פרטיים. לאחר חמש שנים התחתן המורה, ועבר לגור בעיר תל-אביב. בבית המשותף הסתכסך עם שכנו מלמעלה על רקע ניעור שטיחים. הסכסוך הפך לקטטה, ונפתח תיק במשטרה. התיק נסגר, שניהם עשו "סולחה".

עברו עוד חמש שנים, והמורה שלנו רוצה להיות מנהל בית-ספר. הוא מגיש את מועמדותו. פקיד משרד החינוך ממיין את המועמדים בעזרת המחשב, הוא פונה אל בסיס הנתונים המרכזי, ומגלה פרטים "מעניינים" על המורה שלנו: קנסות במס הכנסה, הסתבכות בקטטות, תיק במשטרה. "דמות כזו תהיה מנהל בית-ספר? לא ולא " - אומר הפקיד לעצמו ופוסל את מועמדותו. לא כאן המקום לדון, מהם הקריטריונים לבחירת מנהל בית-ספר, אך כיום, כאשר נבדקים הכישורים הדרושים לתפקיד זה, אין לוקחים בחשבון קטטה מקרית מלפני עשר שנים או תשלומי מס הכנסה באיחור מה, מכיון שהנתונים אינם מרוכזים בשום מקום.

הסמנכ"ל לשעבר של משרד הפנים, הגב' יהודית היבנר, הביאה מקרה היפותתי דומה בהרצאה, שנשאה בכנס איל"א באמצע שנות השבעים. באותה הרצאה הביעה הגב' היבנר את התנגדותה לבסיס נתונים מרכזי של תושבי המדינה. היא גם הוסיפה, שמעטות המדינות בעולם, בהן קיים בסיס נתונים כוללני כזה על התושבים, והכל בגלל צנעת הפרט.

ב-1982 מונתה תת-ועדה של הועדה לחוק ולמשפט, בראשותו של הח"כ דוד גלס. חברי הועדה היו מודעים לסכנות הטמונות בשימוש נרחב בבסיסי נתונים ולהחלפת המידע בין בעליהם. התגבשה הצעה, ובה דרישה למנות רשם ממשלתי להקמת פנקס מרכזי למידע, בו ירוכז כל המידע על התושבים. יחד עם זאת, הודגש הצורך בקביעת הנחיות ברורות וחד-משמעיות לרישום, לעדכון, ולדרכי הטיפול במידע רגיש זה.

פרק 10 - שאלות

1. "בסיסי נתונים מתאימים רק לארגונים או לגופים גדולים מאוד, כמו מדינת ישראל, הסתדרות העובדים וכו'".
הגב על ציטטה זו.

2. הסבר את המושג "רגישות של סגמנט".

3. מהו תפקיד המצביע (POINTER) ב"מבנה השטוח"?

4. מה הכוונה, כאשר אומרים "לסגמנט זה אין נכד"?

5. "בסיס נתונים טבלאי הינו ידידותי למשתמש". מה הכוונה כאן? במה מתבטאת "ידידותי" זו?

6. מנה את היתרונות של השימוש בבסיסי נתונים.

פרק 11 – עיבוד נתונים בתקשורת

במרכז ההדרכה של אחד הבנקים הגדולים, בקורס של פקידי האשנבים, מציגים סרט צבעוני ויפה בשם "דרכו של השיק". סרט זה, שהופק בשנת 1979, מראה את האירועים בבנק במשך לילה אחד, מסגירת הסניפים ועד לפתיחתם למחרת בבוקר.

בכל סניף נארזים לעת ערב המסמכים השונים בחבילות. שיקים לחוד, וטפסים המעידים על הפקדת כסף לחוד. החבילות נשלחות במוניות מכל הסניפים לתל-אביב, אל המרכז, אל המחשב של הבנק. מאילת נשלח החומר במטוס. במרכז ישנן עשרות קלדניות המעלות את החומר על אמצעי קלט (תקליטונים), והחומר מוזן למחשב. תכנית מתאימה בזיכרון מעדכנת את קובצי הלקוחות, ובמקביל מוציאה דו"חות שונים. אחד מהדו"חות נקרא דו"ח היתרות, ובו מופיע כל בעל חשבון עם היתרה החדשה, אותה חישב המחשב על פי כל השיקים וטפסי ההפקדה שהתיחסו לאותו מספר חשבון.

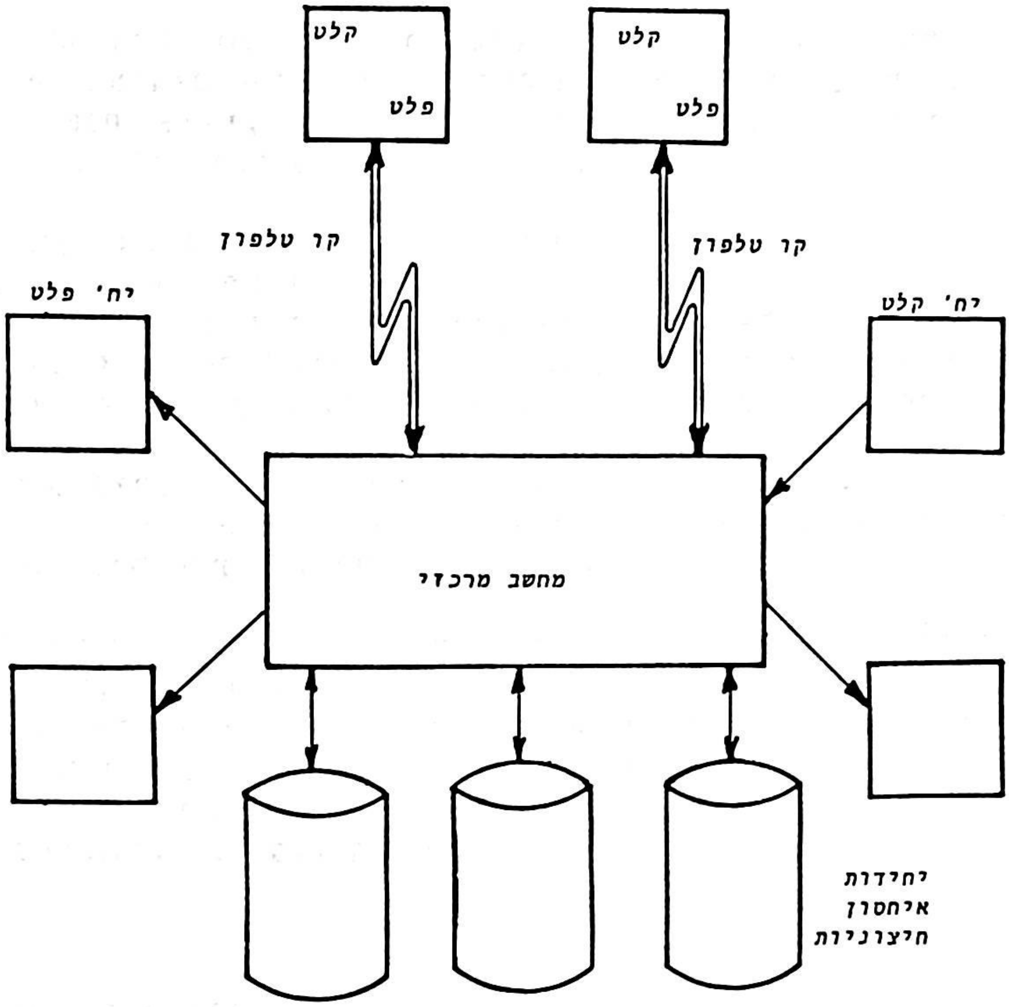
בין השעות שלוש-לארבע לפנות בוקר מופיעה שורה של מוניות ליד פתח הבניין. בתוך דקות נארזות חבילות הדו"חות הממוינות לפי הסניפים, והמוניות יוצאות לדרכן. עד לשעה שמונה בבוקר מגיעים כל הדו"חות לסניפים השונים, ועם פתיחת הדלתות בשעה 08.30 יכול פקיד האשנב לומר לכל לקוח, השואל על כך, מהי יתרת הכסף העומדת לרשותו. עד כאן הסרט.

11.1 עיבוד נתונים מרחוק

כבר לפני שנים אחדות עמדו על העובדה המצערת, שדרך זו של עדכון היתרות בסניפים היא דרך בזבזנית. כמה דלק יכלו לחסוך, אם היו מציבים קוראת תקליטונים ומדפסת בכל סניף, ומחברים אותם אל המחשב המרכזי? קלדניות בתוך הסניף עצמו היו מקישות את פעולות המשיכה וההפקדה. בדרך כלשהי, נניח דרך קווי טלפון, אפשר היה לשגר את הנתונים למחשב, ותוצאות החישוב (היתרות) היו נשלחות בחזרה אל הסניף, שם נמצאת מדפסת, אשר מדפיסה את דו"ח היתרות. מאות ואולי אלפי קילומטרים של נסיעת מונית היו נחסכים מדי יום ביומו בבנק זה, וכמובן גם בבנקים אחרים.

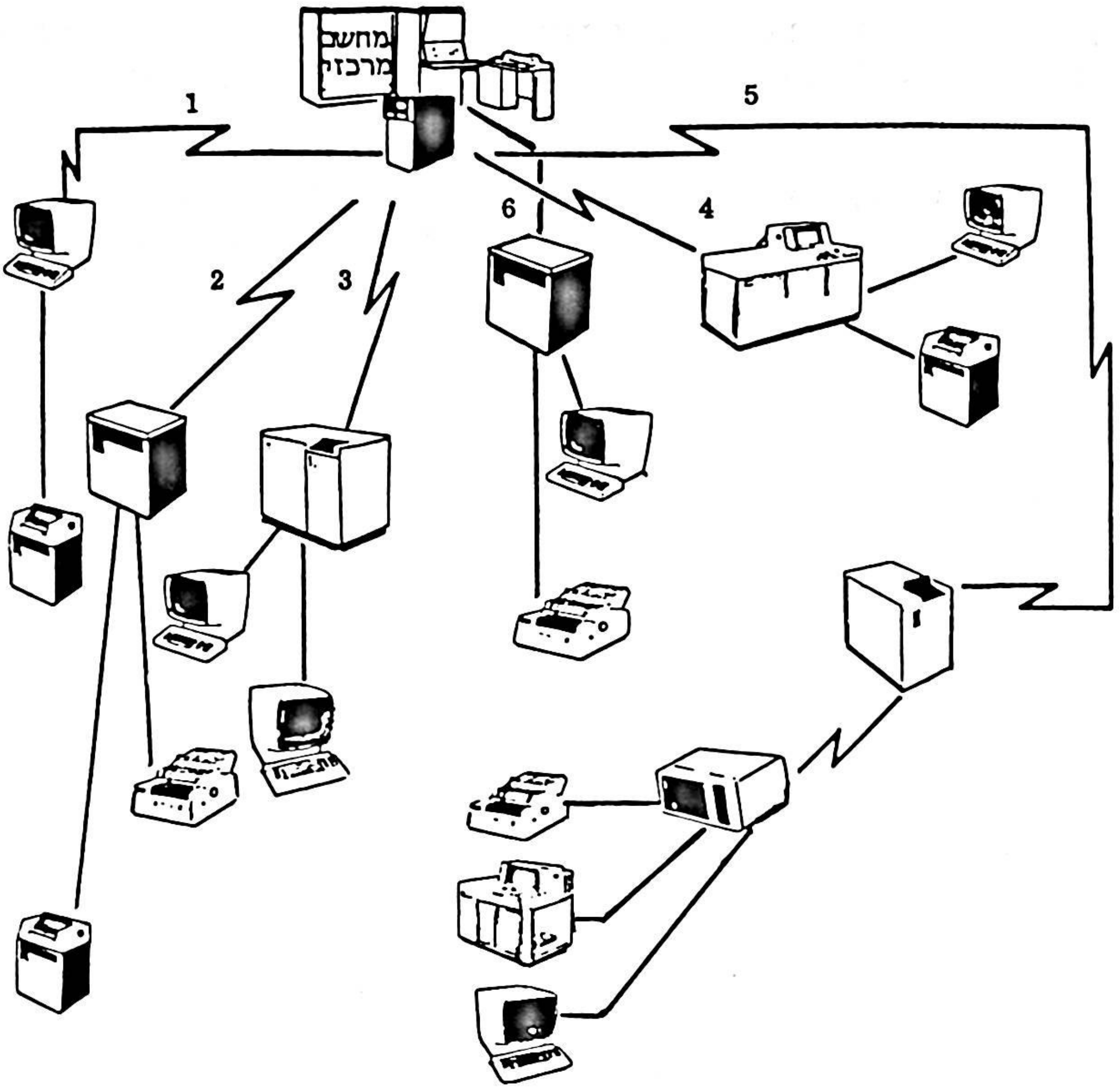
בארה"ב כבר בדקו לפני שנים רבות, האם המחשב יכול לעבד נתונים מרחוק. לקחו יחידות קלט ופלט, וניסו לחבר אותן אל המחשב בכבל ארוך מהרגיל. מפעילי השיטה נתקלו בקשיים מסוימים, אך התגברו עליהם, וכך התברר, שאפשר לחבר יחידות ק/פ ממרחק של מאות

ואפילו אלפי קילומטרים. נמחיש זאת בציור 11-1. מן הראוי לציין כבר עתה, שאין זו הארכה של כבל החיבור שבין היחידה לבין המחשב בלבד. משולבים כאן מתקנים, אשר מאפשרים שימוש בקוי טלפון להעברת הנתונים בין היחידות המרוחקות לבין המחשב.



ציור 11-1 - חיבור יחידות ק/פ מרוחקות

כך הוקמו מערכות <u>לעיבוד נתונים מרחוק - ענ"מ</u> (TELE-PROCESSING), אשר משולבות <u>ברשתות תקשורת</u> (TELECOMMUNICATION NETWORKS). בכל רשת ניתן להבחין בשלושה מרכיבים: המחשב המרכזי, קווי תקשורת ויחידות ק/פ מרוחקות. יחידת קלט/פלט כלשהי, הנמצאת בקצה של קו התקשורת, נקראת <u>מסוף</u> (TERMINAL).



ציור 11-2 - תיאור סכימטי של רשת תקשורת

11.2 סוגי יישומים בענ"מ

בעיבוד נתונים מרחוק ניתן להבחין בכמה יישומים אופייניים, שהתפתחו כתוצאה ממספר גורמים:

- גידול מהיר בהיקף העסקות של מפעלים רבים.
- פיזור גיאוגרפי של מפעלים.
- הצורך בקבלת מידע במהירות מירבית.

.1 איסוף נתונים - DATA ENTRY, DATA COLLECTION

איסוף נתונים בתקשורת מקובל על פי רוב במפעל, שמחלקותיו

פזורות על פני שטח גדול. באולמות הייצור מותקנים מסופים במקומות שונים והפועלים, או מנהל העבודה מדווחים למחשב. נניח שהפועל צריך להרכיב חלק מסוים, אשר מורכב מחמישה פריטים. הפועל ייגש למסוף, ידווח את שעת ההתחלה, מספר פקודת העבודה, מספרים קטלוגיים של הפריטים ואת מספר העובד שלו. בסיום פעולת ההרכבה הוא ידווח על שעת הסיום בלבד.



ציור 3-11 - יח' 3641 לאיסוף נתונים של חברת יבמ

נתונים אלה נאספים במחשב ע"י תכנית מתאימה אשר יוצרת מהם קובץ עבודה. נכונות הנתונים נבדקת, והנתונים הטובים מאוחסנים לעיבוד מאוחר יותר. נתונים, שנמצאו בלתי תקינים - מודפסים בדו"ח שגויים, ועוברים טיפול חוזר.

לעת ערב, לאחר שהעבודה באולמות הייצור הסתיימה, מופעלת תכנית אחרת במחשב, שתפקידה לקרוא את קובץ העבודה עם כל הנתונים, שנאספו במשך היום, לנתח אותם ולבצע בהם פעולות שונות. בתכנית כזאת ניתן לעדכן את מלאי הפריטים במחסן, לעקוב אחר שעות העבודה, לבצע סטטיסטיקות שונות הקשורות בייצור וכו'.

מגוון המסופים לאיסוף נתונים גדול מאד, החל ממסוף פשוט, שהוא יחידת קלט בלבד, ועד למסוף משוכלל בעל יחידת עיבוד זיכרון, ובו תכניות לביצוע בדיקות לוגיות של נתונים לפני שיגורם אל המחשב.

2. שאילתא - INQUIRY

שאילתא הינה פנייה אל המחשב לקבלת מידע. בסוג יישום זה מפנים שאלה אל המחשב, לדוגמה: במחסן מרכזי של בית החרושת ניתן להקיש את מספרו הקטלוגי של פריט מסוים ולשאול, כמה יחידות נמצאות במחסן. התשובה תופיע מיד על המסך. דוגמה אחרת: פקיד בנק מקיש את מספר החשבון של הלקוח על מקלדת המסוף, וכעבור שניות אחדות מוצגת היתרה על המסך.



ציור 4-11 - שאילתא בבנק

יש להדגיש כאן שתי נקודות חשובות:

- השאילתא אינה גורמת שינוי של תוכן הקובץ: כמות הפריטים במחסן או כמות הכסף העומד לרשות הלקוח לא השתנו.

- פרק הזמן, שעבר מרגע הקשת השאלה, ועד רגע קבלת התשובה היה קצר מאד. זהו <u>זמן תגובה</u> (RESPONSE TIME). כיום מצפים מכל רשת תקשורת, שזמן התגובה בה יימשך מספר שניות בלבד.

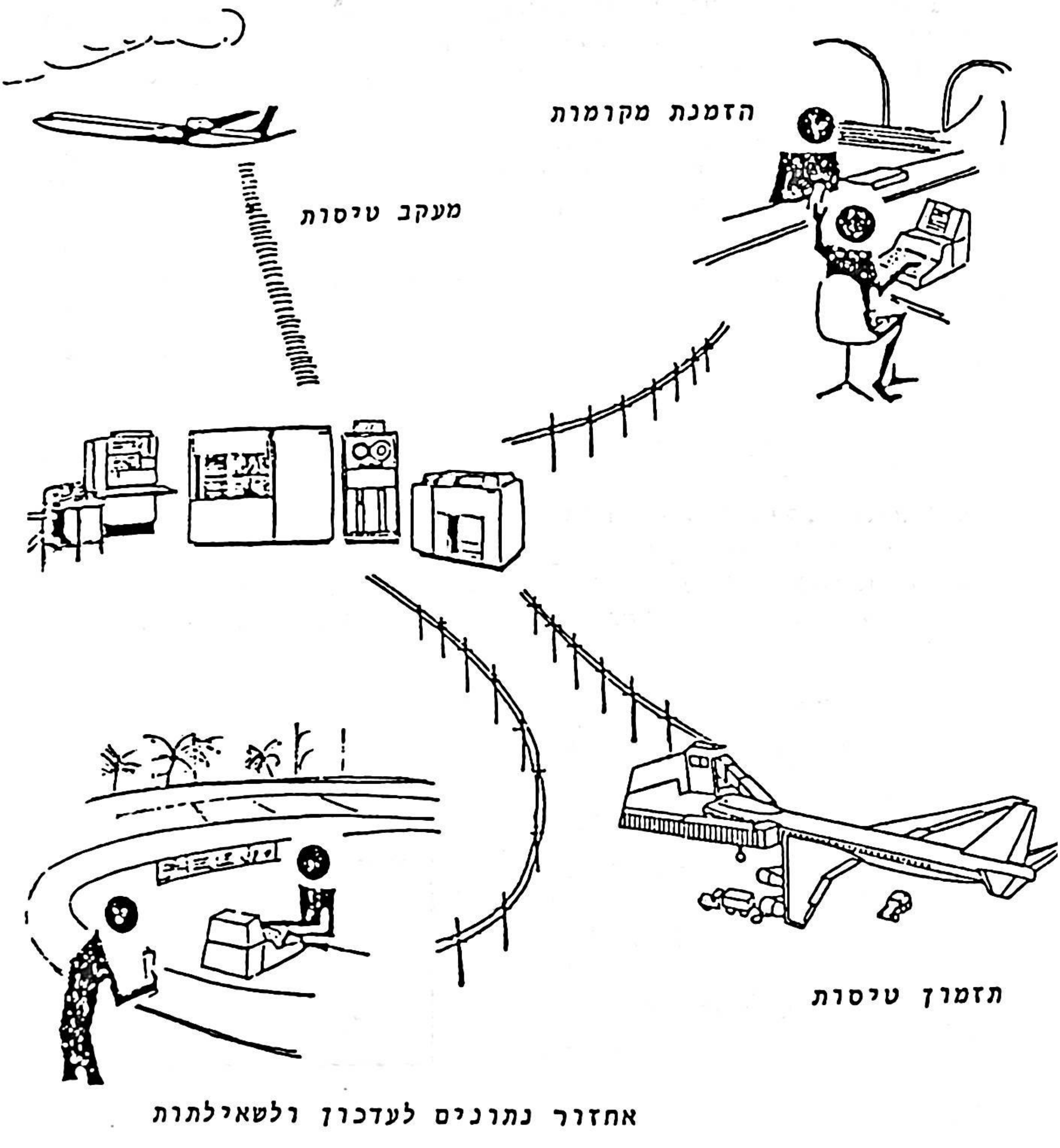
<u>3. עדכון - UPDATE</u>

עדכון הינו פעולה מכוונת לשינוי תוכן הנתונים בקובץ, והוא נעשה לעתים לאחר השאילתא. לקוח בבנק, השואל על יתרת חשבונו, מעונין על-פי-רב למשוך מזומנים. מנהל עבודה, ששאל במחסן על פריט מסוים, יזמין בדרך-כלל את כל הכמות או חלק ממנה.

בפעולת <u>עדכון</u> ישתנה תוכן הרשומה של לקוח בבנק, כיון שאם הוא מושך מזומנים, היתרה תקטן. ובמחסן תקטן הכמות של הפריט, כאשר מנהל העבודה החליט להוציא פריטים מן המחסן לאולם הייצור.

נפנה אל דוגמה נוספת: מר ישראלי רוצה לבלות את חופשתו בלונדון. הוא ניגש למשרד נסיעות של אל-על. הפקידה שואלת באיזה יום מעונין מר ישראלי לנסוע. לאחר קבלת התשובה, תקיש הפקידה את התאריך המבוקש ואת יעד הנסיעה. הפנייה תועבר אל המחשב המרכזי של אל-על, ושם תפנה תכנית מתאימה אל קובץ, שבו רשומות כל הטיסות והמקומות הפנויים בכל טיסה. לאחר קבלת תשובה חיובית מהמחשב, הפקידה תשאל, האם התאריך והשעה נוחים למר ישראלי. אם התשובה חיובית, היא תזמין עבורו מקום באותה טיסה.

הזמנת מקום פירושה <u>עדכון</u> הקובץ, זאת אומרת הורדת מקום אחד מן המקומות הפנוים לאותה טיסה, ורישום שמו של המזמין. תוכן הרשומה בקובץ ומספר המקומות הפנויים השתנה. הקובץ יכיל מעתה פרטים על מר ישראלי, כתובתו, מסי הטלפון שלו ופרטים אחרים.



ציור 5-11 - מערכת שרות בחברת תעופה

4. הזנת עבודות מרחוק - RJE (REMOTE JOB ENTRY)

בשיטת עבודה זו מבצעים עבודות במחשב ממרחקים. המסופים הפועלים בשיטה זו הם לרב דמויי יחידות קלט/פלט של מחשב ולא מסוף לשימוש אישי. בכל מקרה הכינוי מסוף עדיין תקף.

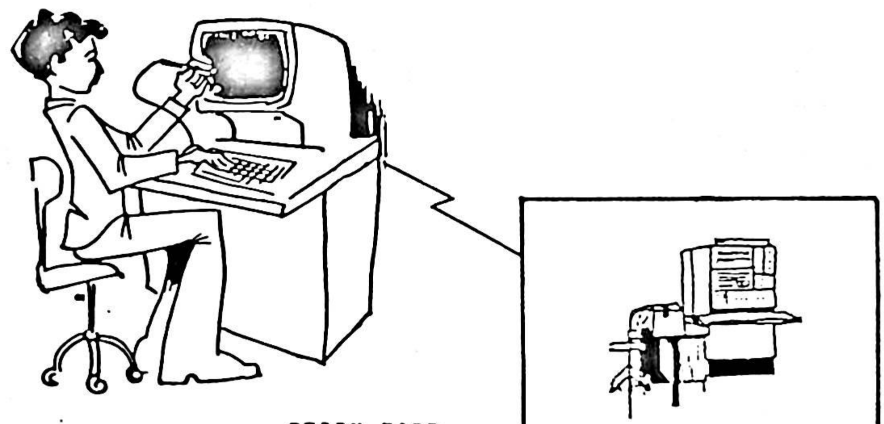
נתאר לעצמנו מוסד כמו אוניברסיטת תל-אביב או הטכניון בחיפה. לפני התקנת המסופים היה כל מרצה או סטודנט צריך לגשת אל בניין המחשב, והיה עליו למסור שם את "עבודתו", כלומר תכנית, אשר כללה הוראות הפעלה וקבצים. העבודות הללו, בין אם דרשו הידור

(קומפילציה), ובין אם היו "מנופות" משגיאות, הורצו במחשב ע"י המפעיל, והפלט של התכנית (בדרך-כלל בצורת דפים מודפסים), נמסרו לבעלי התכניות כעבור שעות רבות, ולעתים למחרת היום.

בשיטת ה-RJE ניתן לשלוח את התכנית מן המסוף אל המחשב המרוחק. המחשב קולט את התכנית, ומכניס אותה לתור לביצוע. בסיום העבודה שולח המחשב את תוצאות העיבוד אל יחידת הפלט המרוחקת, הצמודה ליחידת הקלט, או למסוף האישי. כך ניתן לשלוח תכניות מכל בניין בקריית הטכניון או מכל פקולטה באוניברסיטת תל-אביב אל המחשב. כמובן דרך עבודה זו נוחה וקלה יותר מאשר צורת העיבוד בעבר.

5. הזנת עבודות בהידברות - CONVERSATIONAL RJE

שיטת עבודה בהידברות הינה הרחבה של ה-RJE הרגיל, והיא משרתת בעיקר את תהליך <u>פיתוח היישומים במחשב.</u> השינוי מתבטא בכך, שתכניתן, היושב מול מסוף (לרוב מצג או מכונת כתיבה במקרים נדירים, כיום) יכול לקיים דו-שיח עם המחשב בתוך כדי עבודתו.



ציור 6-11 - פיתוח יישומים בהידברות

תכנית בזיכרון מוכנה לקלוט <u>מסר</u> (MESSAGE) מן המסוף, לפענח אותם, ולשלוח הודעה חזרה אל המסוף. אם נשלחה תכנית להידור, מופיעה כעבור זמן קצר תוצאת התרגום של התכנית על המסך, בציון השגיאות וניתן לתקן אותן מיד ו"להריץ" את התכנית מחדש. במקרה של משלוח תכנית מוכנה לביצוע, יפנה המחשב אל הקבצים המתאימים, יבצע את העבודה, ולפי דרישת התכנית ישלח בסיומה את התוצאות אל המסוף המרוחק. המשתמש יכול לספק נתונים חדשים, לעדכן את הקובץ המרוחק, ולקבל תוצאות מעודכנות.

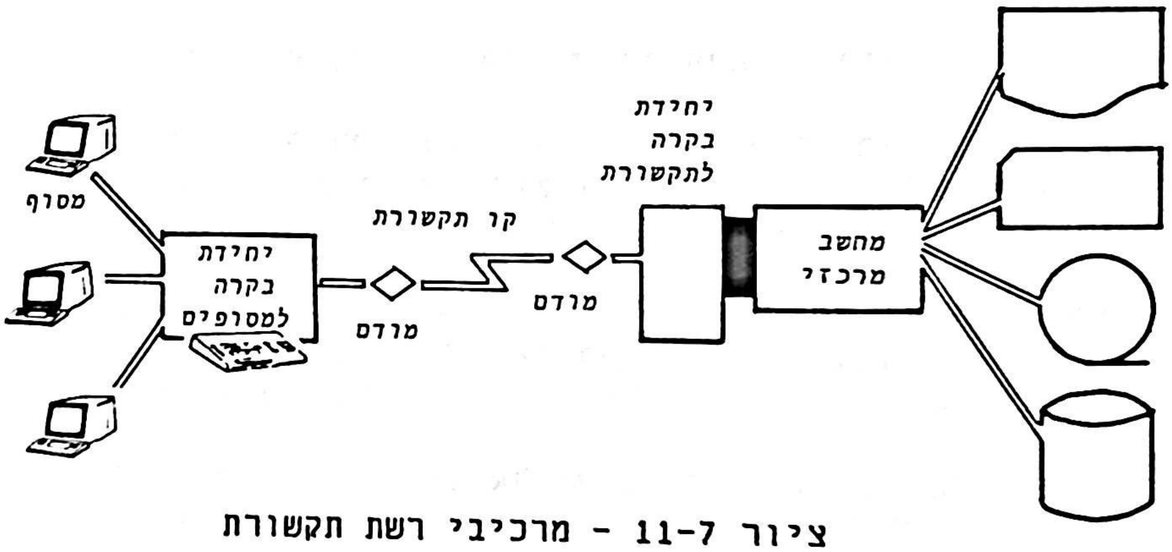
6. ניתוב מסרים - MESSAGE SWITCHING

שיטת ניתוב מסרים מאפשרת להעביר מסר (MESSAGE) ממסוף אחד ברשת אל מסוף אחר, באמצעות המחשב, אשר "מרכז" את עבודת הרשת. כל מסר נשלח עם ציון היעד, שם השולח ופרטים נוספים. במקרה של תקלה במסוף המקבל - יכול המחשב המרכזי, אשר שומר את המסר, לשגר אותו רק לאחר תיקון התקלה. המסוף המשדר יכול לבקש אישור על שידור המסר ללא תקלה.

יש לציין, שבמערכת תקשורת אחת אפשר ורצוי לשלב מספר יישומים שונים, מן הנזכרים לעיל.

11.3 מרכיבי רשת תקשורת

נתבונן בציור 11-7. בצדו הימני של הציור נמצאת מערכת המחשב, המוכרת לנו, ובצדו השמאלי של הציור נמצאים המרכיבים האופייניים של רשת התקשורת. נתחיל את הסקירה מן המסוף.



ציור 11-7 - מרכיבי רשת תקשורת

1. המסוף - TERMINAL

לפי הגדרה, המסוף הינו יחידת קלט/פלט כלשהי הנמצאת בסוף קו תקשורת. מסוף, אם כן, יכול להיות מדפסת, מכונת כתיבה או מסך דמוי טלויזיה ומקלדת. כיום נפוץ השימוש במסופים הנקראים בשם יחידת מצג (DISPLAY UNIT), אשר כולל פרט למסך התצוגה גם מקלדת, בדומה למקלדת של מכונת כתיבה (לשם קלט). החשיבות הגדולה של המצג היא בכך שהוא מאפשר עבודה בהידברות, כלומר "דו-שיח" בין אדם ומחשב.



ציור 8-11 - מצג 3180 של יבמ

למצג מספר יתרונות על פני יחידות קלט/פלט אחרות:

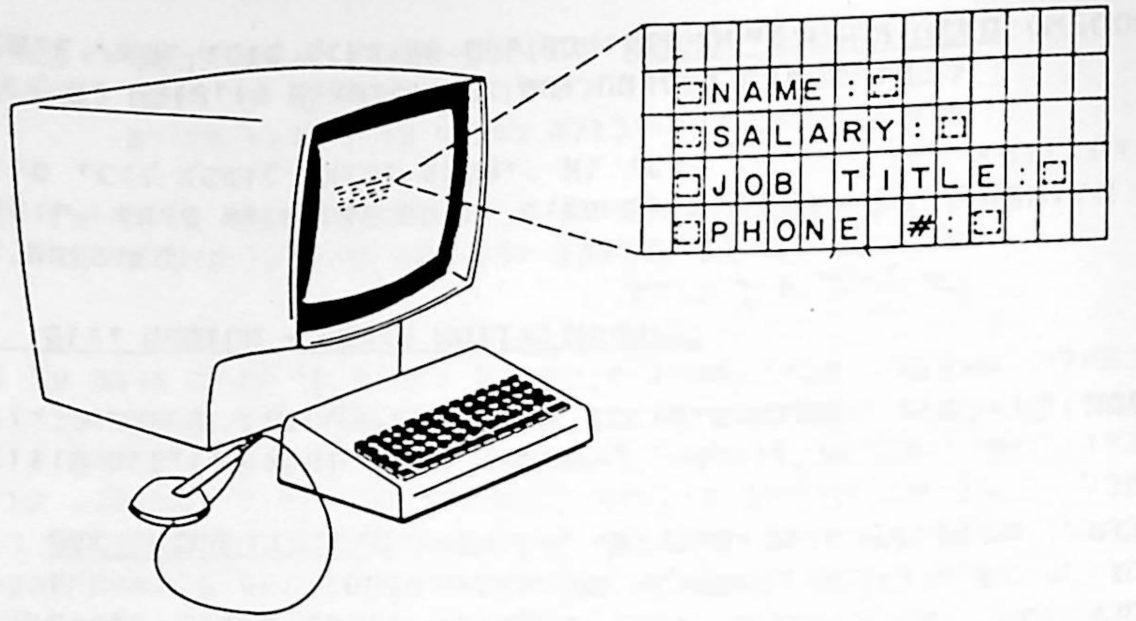
• המצג הינו בעל מאגר (BUFFER) פנימי, אשר יכול לאחסן כל תו הנכתב עליו. קיימים בשוק מצגים, אשר יכולים להציג עד 10,000 סימנים! המשתמש כותב את הנתונים, והם מופיעים הן על המסך והן במאגר. לפני השידור של תוכן המאגר ניתן לעבור על הנתונים, לתקן אותם או להכניס בהם שינויים במידת הצורך. כאשר הכל כשורה, גורמת לחיצה על מקש ENTER לשיגור תוכן המאגר אל המחשב.

• בשעה שפלט המחשב מגיע אל המצג, הוא מוקרן במהירות הרבה יותר גדולה על המסך מאשר קצב ההדפסה על מדפסת, או על מכונת כתיבה.

• ניתן להקרין על המסך צורה של טופס. האדם היושב מול המסך יכול להשלים נתונים במקומות המיועדים לכך, ולאחר מילוי "הטופס" הוא יכול לשדר אותו בחזרה אל המחשב.

דוגמה של טופס ניתנת בציור 9-11.

איסוף נתונים בשיטת הטופס אינו רק מהיר אלא גם מהימן יותר. הסיבה לכך נעוצה בעובדה, שאותה תכנית מחשב, אשר משדרת את הטופס ה"ריק" אל המסך, היא אשר קולטת אותו בחזרה לאחר מילויו. התכנית אינה מאפשרת, לדוגמה, לכתוב נתונים אלפ"אבתיים לתוך שדה נומרי. דוגמה אחרת: אדם נדרש לכתוב

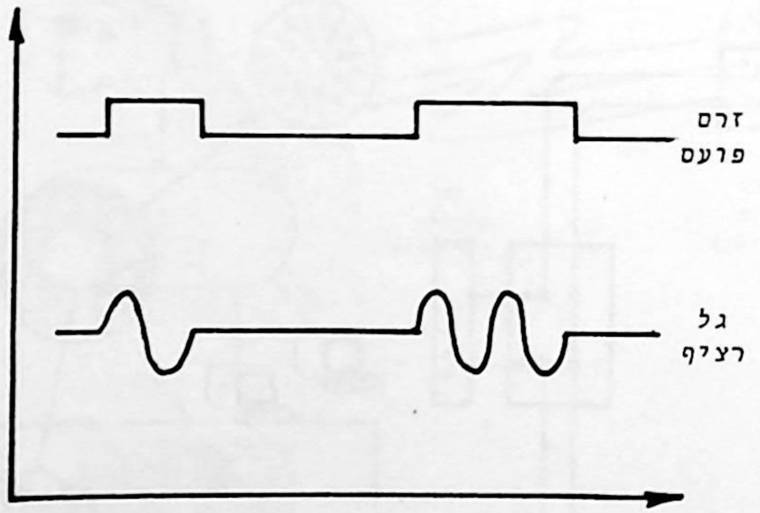


ציור 9-11 - דוגמת טופס "למילוי" על המסך

את מספר תעודת הזהות שלו, המורכב משמונה ספרות. אם מישהו יתקתק פחות משמונה ספרות בשדה של תעודת זהות, תכנית המחשב תעיר על כך, ותבקש מספר "מלא" בן 8 ספרות.

2. <u>מודם - MODEM</u>

הסימנים, המשודרים מן המחשב אל המסוף, או להיפך - אינם מתאימים למשלוח דרך קווי תקשורת או קווי טלפון. אלה הם סימנים ספרתיים (דיגיטליים), אשר מופיעים בצורה של זרם פועם. נוהגים להעביר דרך קווי תקשורת אותות (סיגנלים) בצורה של גלים רצופים (גלים אנלוגיים). הסיבה: גלים אנלוגיים רגישים פחות לעיוותים ולשיבושים בעת מעבר דרך קווים.



ציור 10-11 - זרם פועם

הפיכת האותות הספרתיים של מערכת מחשב לאותות אנלוגיים נקראת **<u>אפנון</u>** (MODULATION), ופעולת ההסבה בסוף הדרך, מאותות אנלוגיים בחזרה לאותות ספרתיים נקראת **<u>גילוי</u>** (DEMODULATION).

מכשיר, אשר יכול לבצע את שתי הפעולות הנ"ל נקרא **מודם** (MODEM) - על פי האותיות הראשונות של שתי המילים האנגליות.

מודם יכול לפעול באופן עצמאי, או יכול להיות בנוי בתוך תיבת המסוף, כחלק אינטגרלי ממנו. מודם חייב להימצא בשני הקצוות של קו התקשורת.

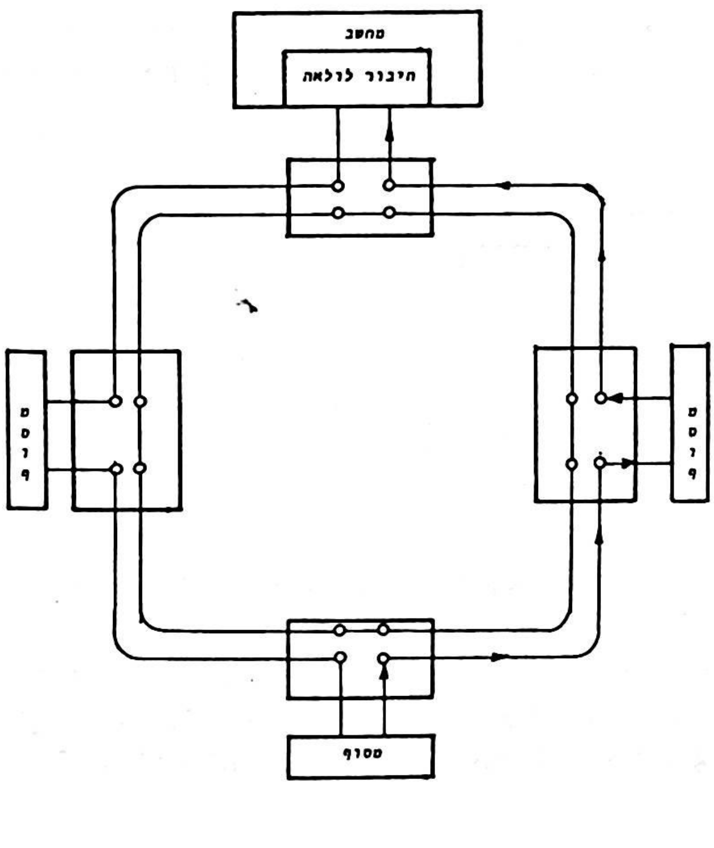
3. קווי תקשורת - COMMUNICATION LINES

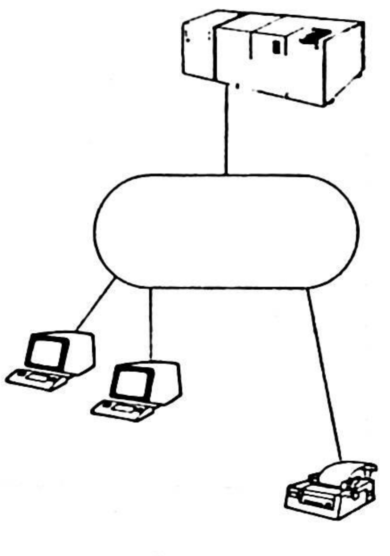
קווי תקשורת נועדו להעביר את הנתונים ברשת. יש להן מספר תכונות פיסיות, אותן נסקור בקצרה:

א. **קצב העברת הנתונים** נקבע ע"י כמות הסיביות שניתנות להעברה דרך אותו קו, בפרק זמן של שניה אחת. קצב זה נקבע, בין היתר, הודות למבנה הפיסי של הקו. קיימות כמה מהירויות סטנדרטיות: 600 סיביות בשניה (BPS), 1200 BPS, 2400 BPS, 4800 BPS, 9600 BPS. אפשר לשדר גם במהירויות גבוהות יותר. קצב העברה של 600 BPS מקביל לקצב העברה של כ-80 תוים בשנייה, וכך הלאה, בהתאמה.

ב. **אופני שידור**, או כיוון העברת הנתונים, מצביעים על דרכים שונות לשימוש בקווי התקשורת. קווי התקשורת נקראים על שם אופן השידור בהם.

- קו **חד-מגמי** (SIMPLEX) מאפשר זרימת נתונים חד-סיטרית. לקו כזה שימוש מוגבל. ניתן להשתמש בו בלולאה (ציור 11-11).





ציור 11-11 · לולאה

- <u>קו דו-מגמי למחצה</u> (HALF-DUPLEX) מאפשר להעביר נתונים לשני הכיוונים אך לא בו-זמנית. קו זה דומה לגשר צר, שרק מכונית אחת יכולה לעבור עליו ברגע מסוים.

- קו <u>דו-מגמי</u> (FULL DUPLEX) מאפשר קשר דו-סיטרי בין שני קצוות הקו באותו זמן. קו זה הינו שימושי מאד, אך גם היקר בין כולם.

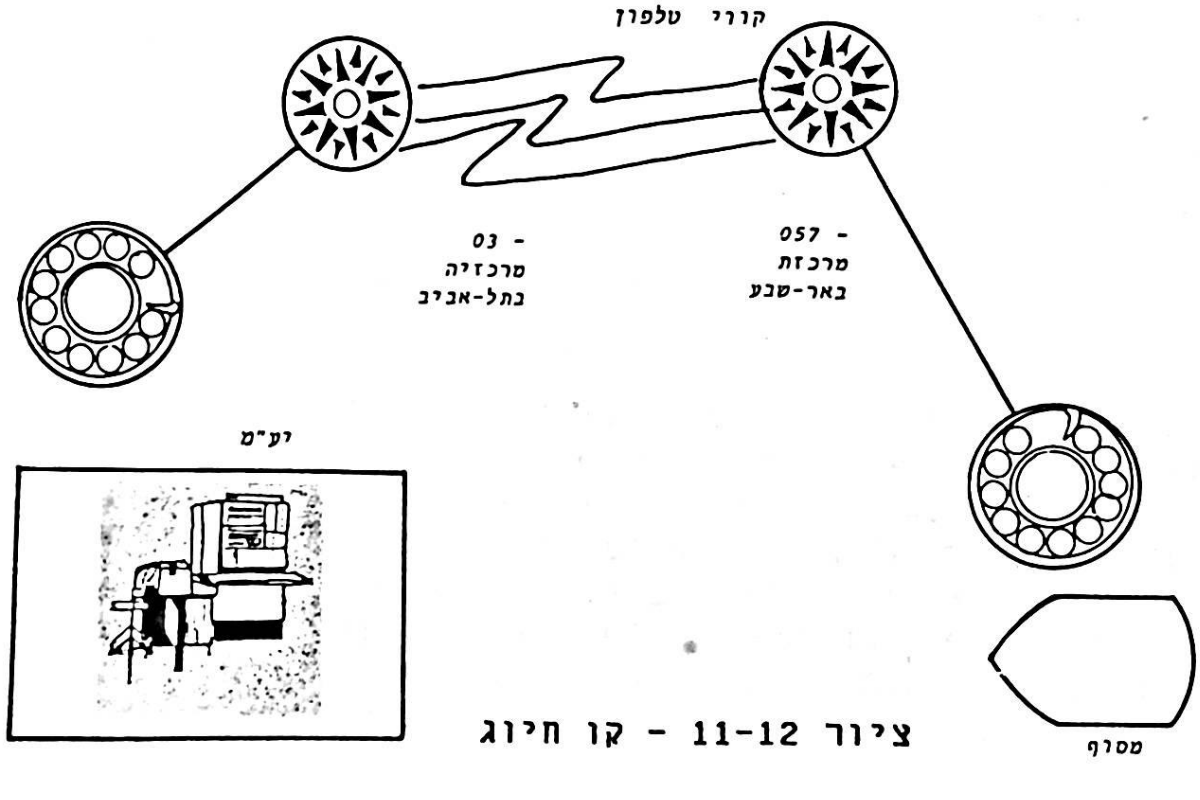
נשאלת השאלה: מנין משיגים קווים כאלה? מי מניח אותם על פני עשרות ומאות קילומטרים? ברב המדינות המפותחות מופקד ענין זה בידי משרד הדואר או משרד התקשורת. בידם יש מלאי גדול של קווים מכל הסוגים, קווים עיליים הנמתחים בין עמודי טלפון, קווים בתוך תעלת בטון תת-קרקעית, או קווים אלחוטיים. בארצות רבות מוכן משרד התקשורת למכור את הקווים או להחכיר אותם לתקופה מסוימת. ישנן ארצות שבהן ניתן זיכיון לחברות פרטיות לפרוש קווי תקשורת.

קיים היבט נוסף: כיצד מחוברים המסופים אל המחשב? ברוב המקומות ניתן לבחור בין שתי צורות של החיבור:

- <u>קווי חיוג או מיתוג</u> (DIAL OR SWITCHED LINES).

- <u>קווים שאינם ממותגים</u> (NON-SWITCHED LINES).

במקרה של שימוש בקווי חיוג נדרשים במחשב ובמסוף מכשירי טלפון לשם חיוג. יצירת הקשר ביניהם נעשית כמו בהתקשרות בטלפון רגיל. הוא יכול להעשות ידנית או אוטומטית, באמצעות מכשיר מתאים. הקשר, שנוצר כך, קיים רק למשך זמן ההתקשרות, ואחר כך נפסק. המנוי משלם רק בעבור זמן ההתקשורת, ולכן סידור זה זול יותר מכל סידור אחר.



ציור 11-12 - קו חיוג

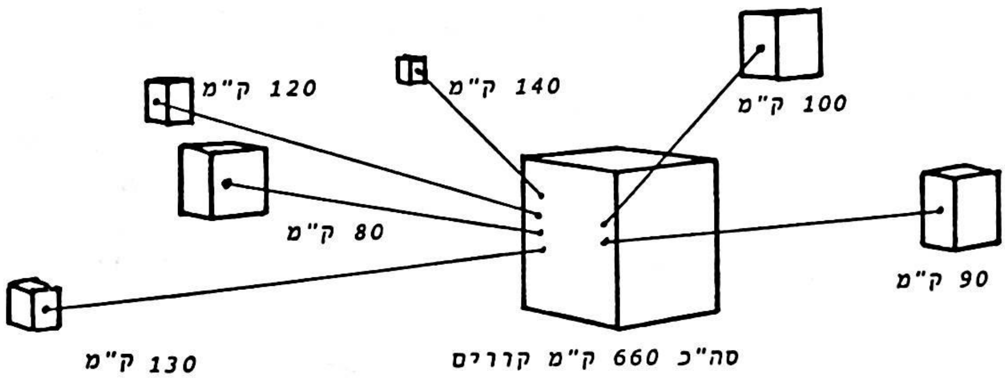
בצד היתרון הכלכלי קיימים חסרונות רבים: הקו תפוס לעתים קרובות, ישנן הפרעות בתוך כדי השידור, העלולות לשבש את הנתונים, המועבר דרך הקווים, אין אפשרות לכייל את הקו, וזה פוגם באיכות השידור, ולבסוף - קיימת אפשרות של "האזנה" (זאת אומרת קליטת החומר המשודר על ידי גורמים זרים, בלתי רצויים). דוגמה של קו חיוג בין מחשב בתל-אביב לבין מסוף בבאר-שבע ניתנת בציור 11-12.

<u>קווים, שאינם קווי חיוג</u> יקרים יותר, אך רב החסרונות של קווי חיוג אינם קיימים בהם.

קווים כאלה ניתן לחבר בכמה צורות:

א. <u>קו נל"ן, נקודה לנקודה - POINT-TO-POINT</u>

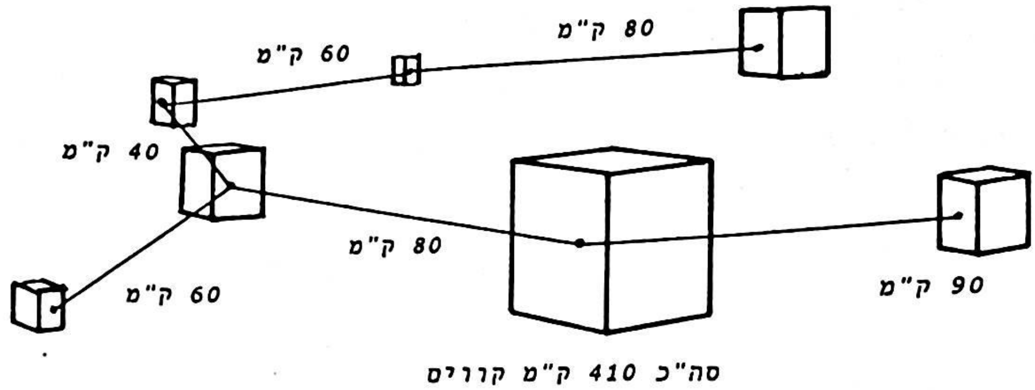
צורת חיבור זו מבטיחה קשר זמין, סודי, ללא הפרעות. ניתן לבחור בקו בעל מהירות מתאימה, דבר, שמנוע מאתנו בקווי חיוג, למשל.



ציור 11-13 - קו נל"ן

ב. <u>קו רב-נקודתי - MULTI-POINT</u>

קו רב-נקודתי מאפשר חיבור של מספר מסופים על אותו קו תקשורת. צורת חיבור זו זולה יותר אך בעלת חסרונות מסויימים: רק מסוף אחד יכול לשדר או לקלוט ביחידת זמן מסוימת, וכל שאר התחנות, על אותה רשת, חייבות להמתין עד שיגיע תורן. עובדה זו מחייבת הנהגת נוהלי עבודה כגון: קביעת תור בין התחנות, מי ישדר ראשון, כמה זמן תשדר כל תחנה, כיצד תדע תחנה, שהגיע תורה לשדר וכו'. ניהול השידור נעשה ע"י תכנת התקשורת אשר נמצאת במחשב.



ציור 11-14 - קו רב-נקודתי

4. יחידת בקרה לתקשורת - COMMUNICATION CONTROL UNIT

סקירת המרכיבים של הרשת מסתיימת בהצגת יחידת הבקרה לתקשורת. היא נמצאת בתווך, בין המחשב לבין שאר מרכיבי הרשת האחרים, ובין המסוף לבין מרכיבי הרשת. ניתן לומר, בצורה אלגורית, שיחידה זו מתווכת בין המחשב לבין העולם החיצוני, ובין המסוף לבין הרשת.

יחידות הבקרה הינן בעלות יחידת עיבוד וזיכרון ולכן ניתנות לתכנות. התכניות, השוכנות בזיכרון יחידת הבקרה, מבצעות תפקידים רבים, שאך לפני עשור שנים היו מוטלים על המחשב עצמו. אולם כדי להבין, מה מוטל על יחידת בקרה, יש לברר קודם משטר קו מהו. לכן, תפקידי יחידת הבקרה יפורטו מאוחר יותר.

11.4 משטר קו

מה יקרה, אם שני האנשים המדברים בטלפון, ידברו בבת-אחת? ציור 11-15 ממחיש זאת היטב. למרות שקיים קשר טלפוני ביניהם, אי-אפשר לדבר על "שיחה...".



ציור 11-15 -
כאשר אין משטר קו...

כמו שאנו נוהגים לשמור על מספר כללים בתוך כדי שיחת טלפון, כגון לא לדבר בו-זמנית עם בן-שיחתנו, או במקרה של אי-הבנה, מבקשים, שיחזור על המשפט, שיאיית מלה כלשהי או בתוך כדי האזנה לאשר: "כן, כן..." - כך יש צורך בנהלים קבועים, כדי להסדיר את תנועת הנתונים דרך קווי התקשורת. הנהלים נקראים "פרוטוקול" או <u>משטר קו</u> (LINE CONTROL). הם מתייחסים להיבטים שונים של השידור ברשת: מי ישדר? מתי? אל מי מיועדת התשדורת? האם הקווים פנויים? הבה נבחן בקצרה את הכללים העיקריים:

1. <u>סריקה לקראת שידור</u>

ניתן לסרוק את הרשת לקראת השידור באחת משתי הדרכים:

א. <u>חקירה - POLLING</u>

המחשב פונה אל כל אחד מן המסופים בשאלה, האם יש לו נתונים לשדר. אם התשובה חיובית, המסוף יכול לשדר, ואם התשובה שלילית, השאלה עוברת אל המסוף הבא.

סדר הסריקה נקבע בטבלה מיוחדת, בה רשומים כל המסופים. בזמן בניית הטבלה ניתן לתת עדיפות למסוף מסוים או לקבוע שאל מסוף א' יפנו פי ארבעה יותר, מאשר אל מסוף ב' ופי שניים יותר מאשר אל מסוף ג', למשל.

ב. <u>מעינה - ADDRESSING או בחירה - SELECTION</u>

מעינה מתאימה למקרה בו המחשב רוצה לשדר אל מסוף מסוים או אל כמה מסופים. כאן המחשב "כאילו" שואל את המסוף, "האם אתה מוכן לקליטה?" אם התשובה חיובית - מתחיל המחשב לשדר.

המחשב יכול לשדר אל מסוף בודד או אל קבוצת מסופים או אל כל המסופים ברשת. זה נקבע בנהלים, כמובן. כמו כן קיים נוהל למקרה, שהמסוף משיב תשובה שלילית, או אינו משיב כלל.

לסיכום, החקירה מתאימה לקלט אל המחשב, ומעינה - לפלט מהמחשב.

2. <u>שיטות תמסורת</u>

דרך קווי התקשורת ניתן לשדר בשני אופנים:

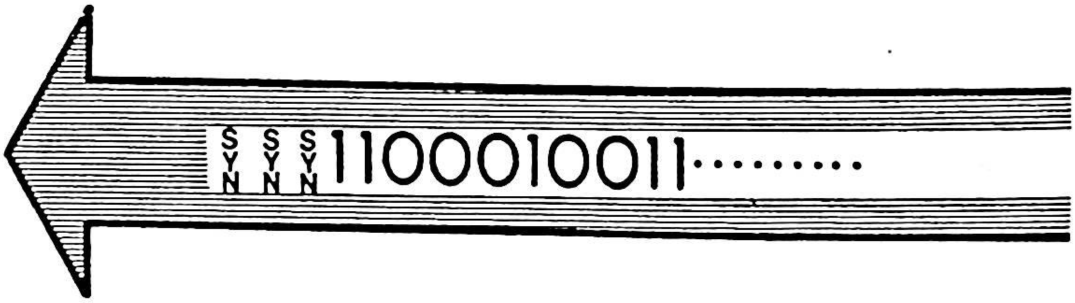
א. שיטה אסינכרונית

ב. שיטה סינכרונית

בשיטת תמסורת **אסינכרונית** (ASYNCHRONOUS), שהיתה נהוגה כבר לפני עשרות שנים, לכל תו משודר יש להקצות סיבית התחלתית (START BIT), ובסוף התו יש להוסיף סיבית, המציינת את סוף התו (STOP BIT). הצורך בהוספת סיביות נבע מן העובדה, שהיו הפרעות רבות בקווים ולעתים התחנה המקבלת לא הבחינה, אם האות החשמלי הנקלט הינו תו או הפרעה, שנגרמה בגלל גורם חיצוני (ברק, טרנספורמטור, מטוס וכוי). שיטה זו נקראת גם **START/STOP** (S/S) בגלל הסיביות שנתוספו לתווים. השיטה התאימה במיוחד לרשתות של שנות ה-60 וה-70, כאשר הקווים היו בעלי מהירות נמוכה, והמסופים היו פשוטים וזולים יחסית.

בשיטה **הסינכרונית** (SYNCHRONOUS) נשלחים בתחילת השידור תווים מיוחדים מן התחנה המשדרת אל התחנה הקולטת. תווים מיוחדים אלה באים לתאם בין שתי התחנות ועל-פי-רב להפעיל קוצב זמן מיוחד. לאחר מכן יבוא השדר ברצף אחד, גם אם הוא ארוך מאד. אין צורך בהוספת סיביות START/STOP, וזה כמובן מגדיל את כמות התווים, העוברים דרך קווי התקשורת ביחידת זמן.

קיימים שני משטרי קו סינכרוניים, אשר נתאר אותם בהמשך, מכיון שמשטרי קו אלו הם הנפוצים כיום ברב רשתות התקשורת.



ציור 16-11 - הודעה עם תוי סינכרון בראש

3. BSC - BINARY SYNCHRONOUS COMMUNICATION

משטר קו BSC, כפי ששמו מרמז, הינו אוסף של נהלים, אשר מאפשרים שידור סינכרוני של נתונים. ברשת תקשורת, הפועלת תחת משטר קו כזה, ניתן להפעיל מסופים מסוגים שונים (מדפסות, מצגים, וכוי) תחת מגבלות מסויימות. ברשת ניתן לכלול קבוצות קווים בעלות מהירויות בינונית וגבוהה.

במשטר קו BSC קיימות שתי דרכים כדי **לבדוק את שלימות הנתונים** המועברים ברשת:

- **שיטת CRC**. בשיטה זו נערך, בדרך מיוחדת, סיכום של כל התווים וגושי הנתונים המשודרים. התוצאה נכתבת בשני בתים בסוף כל גוש, ונשלחת יחד איתו.

התחנה המקבלת בודקת את התשדורת המתקבלת, ובונה לעצמה שני בתים משלה לשמירת התוצאות. לאחר מכן נעשית השוואה בין זוגות הבתים, ואם מתגלה אי-שוויון ביניהם, תודיע התחנה הקולטת על שגיאה לתחנה המשדרת. במקרים כאלה נוהגים לחזור על שידור הגוש הזה פעם נוספת, או אפילו פעמים נוספות.

• שיטת האישורים החיוביים. שיטה זו הונהגה כדי למנוע אפשרות, בה גוש שלם של שידור הולך לאיבוד. לאחר שגוש נקלט בסדר נשלח אישור לתחנה המשדרת, ואז היא יכולה להמשיך בשידור הגוש הבא. אם אין אישור חיובי לקליטה - משודר גוש הנתונים שנית.

במקרים של תקלה, או שיבוש בקליטה ממסוף, המחשב יכול וגם צריך לנעול את מקלדת המסוף. בכך הוא מונע הקשה של נתונים חדשים כל עוד הוא אינו יכול לאשר את הקליטה של נתונים קודמים.

בין האפשרויות של שידור בקו BSC יש להזכיר את אפשרות השידור של נתונים בינריים, דבר שלא ניתן לביצוע בקו START-STOP.

4. SDLC - SYNCHRONOUS DATA LINK CONTROL

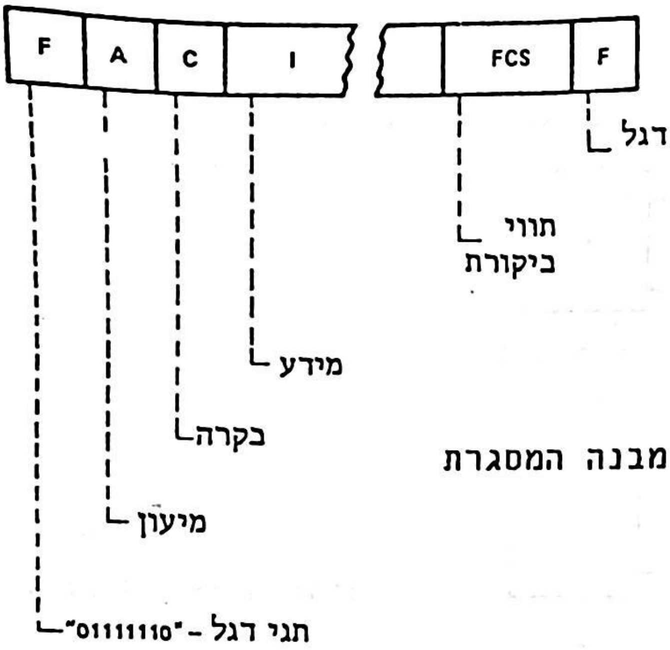
משטר קו SDLC מטפל בכל התחומים שטיפלו משטרי הקו האחרים שתוארו, דהיינו חקירה, מעינה, בדיקת תקינות וטיפול בתקשורת, אולם הוא משוכלל יותר במספר תחומים.

נציין כאן בקצרה את המיוחד שבמשטר קו זה:

א. מבנה המסגרות. התשדורות נשלחות במבנה קבוע, הנקרא מסגרת (FRAME). המסגרת מחולקת ל-3 חלקים: בהתחלת המסגרת כותרת בת 3 תווי בקרה, במרכז - קטע הכולל את הנתונים, ובסוף שוב קטע בן 3 תווים, לבקרה. תווי הבקרה כוללים, בין השאר, את כתובת היעד.

ב. רשת התקשורת בנויה מצמתים (NODES), שביניהם ישנם אפיקי התקשורת (COMMUNICATION CHANNELS) הנקראים חוליות (LINKS). בכל חוליה מוגדר צומת ראשי (PRIMARY NODE) וצומת משני (SECONDARY NODE).

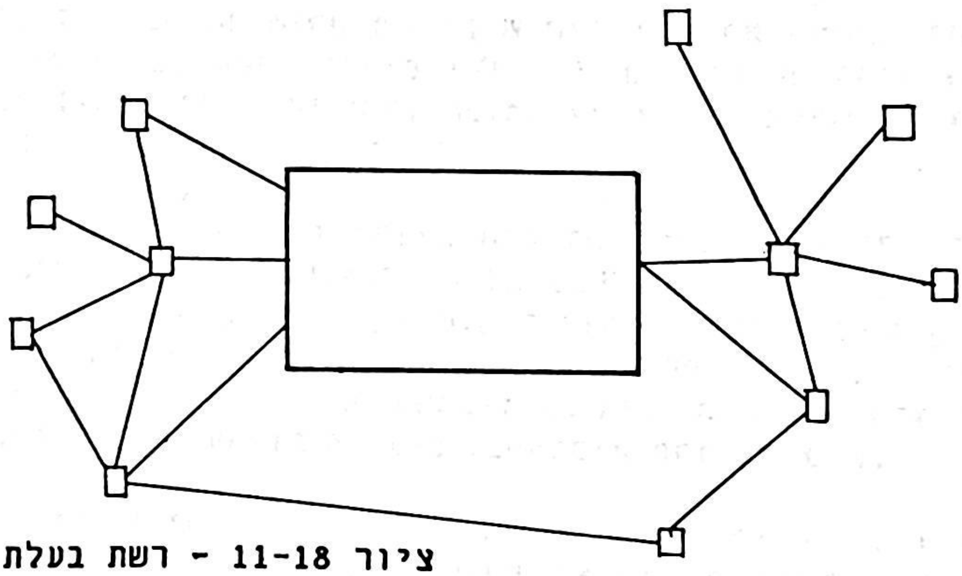
המסרים משודרים דרך הצמתים אל היעדים השונים. תכנית מתוחכמת בודקת את העומס על הקווים, ומנתבת את המסרים דרך צמתים פחות עמוסים. כל מסר נשלח בליווי תווי פיקוח מיוחדים מצומת לצומת. הדבר דומה למשלוח מכתב



ציור 11-17 - מבנה המסגרת

במעטפה. על המעטפה נכתבות שתי הכתובות, הכתובת הסופית של היעד וכתובת הצומת הקרוב. בכל צומת, בהגיע "המכתב" נבדקת תקינות הקליטה, ונרשמת כתובת חדשה בעבור הצומת הבא. הכתובת הסופית אינה משתנית. במקרה של תקלה נשלחת הודעה אל הצומת השולח, שם עדיין נשאר "המקור" וניתן לשדר את המסר פעם נוספת, או לנסות להעביר אותו בצמתים אחרים.

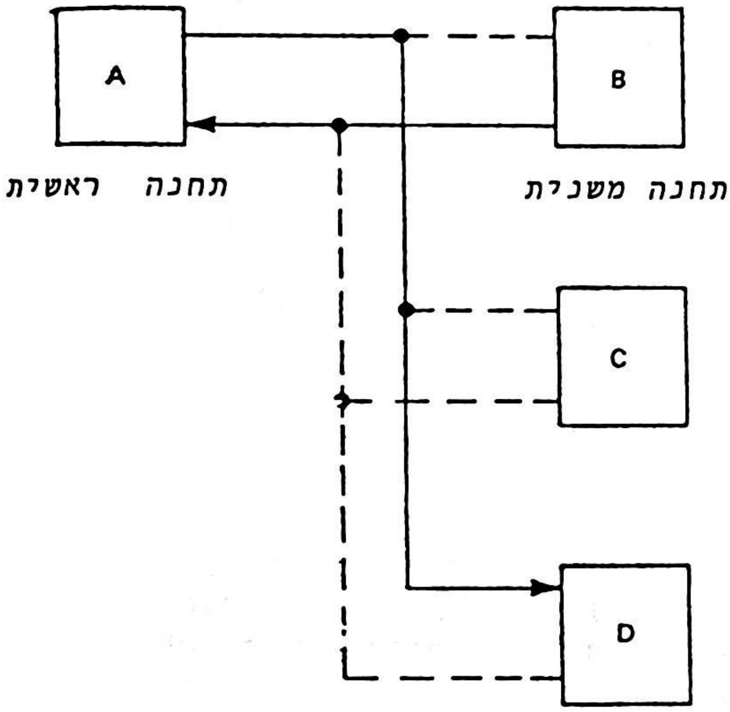
משטר קו SDLC מחייב קיום מספר תנאים: דרושה יכולת עיבוד מסוימת ויכולת אגירה במסופים וביחידות הבקרה. המסופים ויחידות הבקרה החדשים הותאמו לדרישות אלו.



ציור 11-18 - רשת בעלת צמתים

בנוסף לכך משטר קו זה מאפשר חיבור מסופים מסוגים שונים (מדפסת איטית ומצג מהיר) על אותו קו. ניתן להפעיל כל תכנית יישום מכל מסוף שהוא (אפשרויות אלו לא היו קיימות

ב-BSC). בעזרת השימוש בקווים דו-מגמיים (FULL-DUPLEX ניתן לשדר על אותו קו בכיוון אחד ולקלוט בכיוון הנגדי זה מביא, כמובן, לחיסכון בזמן התקשורת.



ציור 11-19 -
A משדר ל-D ובאותו
זמן "מקשיב" ל-B

11.5 יחידת בקרה לתקשורת

הזכרנו קודם פעולות שונות אשר נעשות ברשת התקשורת בין שתי תחנות, בין מסוף לבין מחשב. המחשב תואר לצורך זה כתיבה אחת, ללא אבחנה של הרכיבים השונים התורמים לקליטה ולשידור של המסרים.

הטיפול ברשת התקשורת מחייב שיהיו במערכת המחשב יחידות, או אביזרים, אשר יודעים "לתווך" בין הרשת לבין היע"מ. נוכל לומר שנדרש "שוער", העומד לפני שער המחשב, בודק כל נכנס ויוצא, מתאם ומפקח על הנעשה.

ברב מערכות המחשבים קרויה היחידה המתווכת - <u>יחידה בקרה לתקשורת</u> (COMMUNICATION CONTROL UNIT), שאליה קשורות יחידות קלט/פלט רבות - המסופים. יחידת הבקרה כוללת יחידת עיבוד וזיכרון ותכנה מיוחדת המותאמת לביצוע התפקידים המוטלים עליה. חלוקת התפקידים בין המחשב לבין יחידת הבקרה שונה במערכות מחשבים ובמערכות תכנה שונות.

במחשבים זעירים וקטנים, ולעתים גם במחשבים בינוניים, נמצא את יחידת הבקרה לתקשורת כאביזר המורכב בתוך תיבת המחשב יחד עם שאר הרכיבים, והמשימות מתחלקות בין יחידת הבקרה לבין היע"מ והתכנה שבזיכרון. במחשבים בינוניים וגדולים יחידת הבקרה מותקנת בתיבה נפרדת והיא גם עצמאית מבחינת ביצוע המשימות, עם תלות קטנה ביע"מ.

מהירות הביצוע של המחשב גבוהה בסדרי גודל ממהירות השידור בקווים. הקווים המתחברים למחשב יכולים להיות בעלי תכונות שונות מבחינת מהירות השידור, משטרי קו, סוג הקו, צורת החיבור של המודם, ועוד. יחידת הבקרה לתקשורת מהווה מישק בין העולם החיצוני של קווים ומסופים לבין המחשב.

על יחידת הבקרה מוטלים תפקידים רבים, שהחשוב בהם הוא ניהול משטר הקו וביצוע סריקה ומעינה ותיווך בין הקליטה והשידור של מסרים שלמים מן המחשב לבין קליטה ושידור לקווים של תווים או גושים עם תווי בקרה. יחידת הבקרה בודקת את שלימות התשדורות על פי תווי הבקרה, או מוסיפה תווי בקרה בעת שידור ומטפלת במקרי תקלה. היא יכולה לחייג ולקלוט חיוג בקו חיוג. היא יכולה לבצע בדיקות של קווים, מודמים ומסופים ולהתריע על תקלות.

אפשר לחבר שתי יחידות בקרה זו לזו כדי לווסת ביניהן את עומס הרשת, כלומר את התעבורה שבין המסופים והמחשב.

כל הפעולות האלה, ואחרות, מתבצעות ביחידת הבקרה באמצעות רכיבי החמרה והתכנה שלה. הקשר התפקודי בין המחשב לבין יחידת הבקרה מתקיים באמצעות תכנה לניהול תקשורת אשר נמצאת במחשב.

<u>11.5.1 יחידת בקרה למסופים</u>

ניהול התקשורת בין המחשב לבין המסוף נמצא בשליטה של שתי יחידות בקרה. האחת נמצאת בצד המחשב, ובה דנו זה עתה, והשניה נמצאת בצד המסוף, וקרויה <u>יחידת הבקרה למסופים</u> (TERMINAL CONTROL UNIT). היא ממלאת תפקידים דומים לתפקידי יחידת הבקרה לתקשורת: שליטה במסוף, ניהול התקשורת אל המחשב, והעברת תשדורות בין המאגר שלה לבין מאגר המסוף.

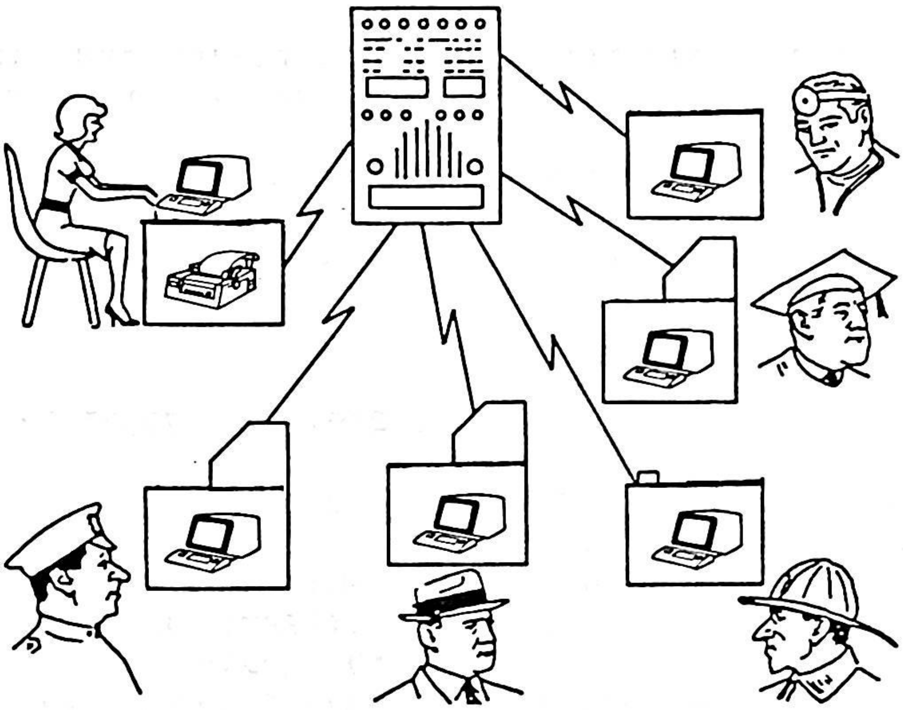
כדי שמסוף יוכל "לדבר" עם המחשב, עליו להכיל יחידת עיבוד אשר תוכל לקיים את משטר הקו. כאשר יחידת הבקרה מותקנת בתוך תיבת המסוף לפנינו <u>מסוף עצמאי</u> (STAND ALONE), וכאשר מסופים אחרים (מצגים ומדפסות) מקבלים שירות מיחידת בקרה אחת לפנינו <u>מקבץ מסופים</u> (CLUSTER). שיתוף המסופים בשירותי יחידת בקרה אחת מאפשר להוזיל את המחיר של המסופים הבודדים.

את המסופים אפשר לחבר ליחידת הבקרה, לרב בכבלים של עד 2,000 מ'. כאשר רוצים להתקין מסופים בסביבת המחשב, אפשר לחבר את יחידת הבקרה למסופים באופן ישיר למחשב, ללא קו תקשורת. זהו <u>חיבור מקומי</u> (LOCAL) לעומת <u>חיבור בתקשורת</u> (REMOTE). הוא מאפשר קצב העברה של עשרות אלפי תווים

בשנייה לעומת קצב העברה של מאות תווים בשנייה, בקווי תקשורת.

11.6 תכנה לניהול תקשורת

רשת התקשורת עם עשרות או מאות המסופים שבה מסמלת את הפריצה של המחשב מתוך מתקן המחשב אל שולחנות העבודה במשרדים, מבני-תעשיה, אולמות הבנקים ועוד. אין עוד צורך "לבוא אל המחשב", "להביא את החומר למחשב" - המחשב בא אל העובד, אל המשתמש, באשר הוא שם.



ציור 11-20 - המחשב לשרותו של כל אחד ואחד

אולם פריצה זו מביאה עמה בעיות רבות, חדשות, שבעיבוד המסורתי, בשיטת האצווה - לא חשבו עליהן. במקום 10-20 יחידות קלט/פלט מחברים אל המחשב מאות מסופים, וכל מסוף כזה מופעל על ידי מישהו, אשר ייתכן וכלל לא למד על מחשבים.

נניח שציור 11-20 מציג לנו עובדי אוניברסיטה מסוימת. פרופסורים לרפואה ולמשפטים, מרצים לכלכלה ולהנדסה מפעילים את המחשב בצד הפקידות במזכירות הסטודנטים אשר מטפלות בציונים. כל אחד רוצה להפעיל תכנית אחרת במחשב, וכל אחד מעונין שהמחשב "יענה" לו מיד! בנוסף לכך יש לקחת בחשבון, שכל תכנית עובדת עם קובץ משלה או עם כמה קבצים.

האם המחשב יכול לענות למאות פניות בבת-אחת? מה יהיה זמן התגובה? מה יקרה אם שתי תכניות פונות אל אותו קובץ או אל אותה רשומה?

כשם שנפתרו בעיות דומות בשיטת התכנות המרובב ע"י מערכת ההפעלה, כך גם נמצאו התשובות הנאותות לשאלות הללו בעיבוד נתונים בתקשורת. תכניות הפיקוח ותכניות שירות של מערכות ההפעלה, מנהלות את העבודות ואת הנתונים, מטפלות בהקצאת הזיכרון וכדומה. תכניות מיוחדות בתחום תקשורת הנתונים, נכתבו כדי לטפל בכל אותם מצבים מיוחדים, שנוצרו בגלל תנאי העבודה המיוחדים, אשר מאפיינים רשת תקשורת.

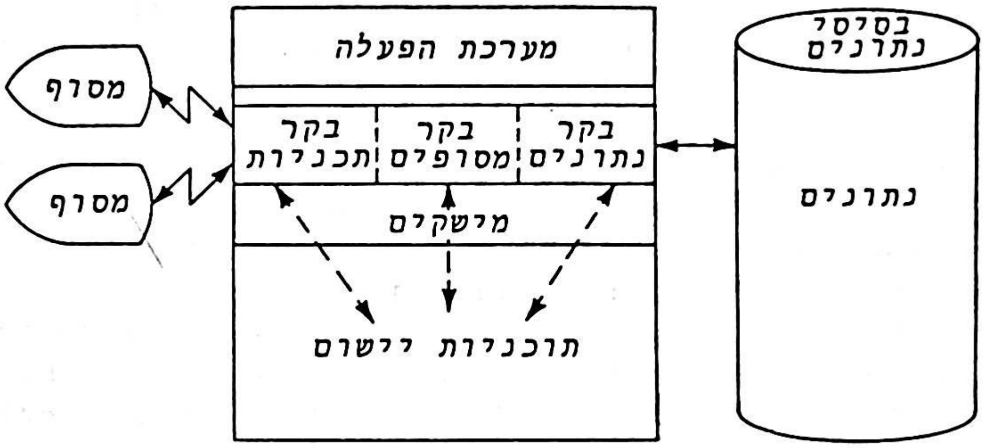
באופן כללי ניתן לראות תוכנה כזאת כאוסף תכניות, שתפקידן המרכזי לנהל רשת תקשורת בצורה היעילה ביותר. כדי להשיג את המטרה הזאת על התכנה לטפל בתחומים הבאים:

- ניהול ופיקוח של רשת התקשורת
- טיפול בתורים
- ניהול תכניות העיבוד
- טיפול בקבצים
- טיפול בתקלות

11.6.1 תכניות לניהול תקשורת

נתבונן בציור 11-21, ונתאר לעצמנו, שמאות מסופים של בנק מסוים מתחברים אל המחשב. בזיכרון המחשב נמצאת תכנה מיוחדת לעיבוד נתונים בתקשורת: תכנית בקרה לתקשורת (COMMUNICATION CONTROL PROGRAM) אשר מורכבת משלשה חלקים עיקריים.

א. בקר מסופים יטפל בכל הפניות המגיעות מן "העולם החיצוני", מן המסופים אל המחשב. פניות אלה חייבות להיבדק תחילה. טבלאות מיוחדות שומרות בזיכרון את השמות של כל המסופים, והן את שמות האנשים הרשאים להפעיל אותם. כל אדם המפעיל מסוף נשאל על ידי מנהל התקשורת, מהי הססמה שלו וזו נבדקת מול רשימת הסיסמאות אשר רשומות בטבלה בזיכרון, או בדיסק.



ציור 11-21 - מרכיבי מערכת DB/DC

ב. <u>בקר תכניות</u>, מטפל בכל תכניות העיבוד, שניתן להפעיל דרך המסופים, כגון תכנית למשיכת מזומנים, לבדיקת היתרה ועוד.

ג. <u>בקר נתונים</u>, ישלוט על כל הפעולות הקשורות בקריאה, כתיבה ועדכון הנתונים בתוך הקבצים השונים של הבנק.

נתאר את תהליך העיבוד בתקשורת באמצעות דוגמה:

נניח שנערה ניגשת אל המסוף, הקשור למחשב של הבנק, ומקישה את הססמה (או את מספרה הסודי). בקר התקשורת בודק את נכונות הססמה ואח"כ שואל מה הפעולה המבוקשת. הנערה מבקשת לדעת, מהי היתרה בחשבונה. הבקשה מועברת דרך בקר התקשורת והוא מסמן בטבלאותיו מאיזה מסוף באה השאלה. הבקשה מועברת עתה אל בקר תכניות הביצוע. התכניות השונות של הבנק, כגון תכניות עובר ושב, תכניות מטבע חוץ וכו' נמצאות בספריית התכניות בדיסק. בקר התכניות מעתיק את התכנית לבדיקת יתרה אל הזיכרון, בתוך כדי סימונים מתאימים בטבלאות שונות.

התכנית מתחילה לעבוד, ומבקשת לקרוא את רשומת הנתונים. לשם כך מועברת הבקשה אל בקר הנתונים, אשר בודק אם רשומת הנערה בכלל קיימת בקובץ הנתונים. ואם היא קיימת, האם היא מותרת לקריאה או שמא תכנית כלשהי מסניף בנק אחר מטפלת בה. אם הרשומה "פנויה", כלומר, שאין תכנית אחרת אשר משתמשת בה באותו רגע, בקר הנתונים מעתיק את הרשומה מן הדיסק אל הזיכרון והיא הופכת ל"תפוסה". התכנית מוצאת ברשומה את היתרה ומעבירה אותה, יחד עם הודעה מתאימה, אל בקר התקשורת. בקר התקשורת בודק בטבלאות לאן מיועדת ההודעה, ואם קו התקשורת פנוי, ושולח את ההודעה אל המסוף, שם ממתינה הנערה לתשובה.

פרק הזמן, העובר מרגע הקשת השאלה ועד קבלת היתרה על המסוף, נקרא, כזכור, זמן תגובה, והוא נמשך 3 עד 5 שניות בלבד!

הדוגמה הזאת הראתה, מהן הפעילויות, שנתבצעו ע"י שלושת החלקים של תכנית התקשורת במקרה של פנייה אחת ממסוף אחד. עתה נתאר לעצמנו, שמגיעות בעת ובעונה אחת מאה או מאתיים פניות מן המסופים. התכנה לניהול תקשורת חייבת להיות ערוכה גם למצב כזה, ולטפל בתורים, שעלולים להיווצר. חלק מהפניות יזכה לשירות מידי, ואילו חלק אחר ייכתב לדיסק מגנטי, שעליו בונה המערכת תור של פניות ממתינות. ברב המערכות נהוג לשלוח הודעת "הרגעה" אל המסוף בזו הלשון: "הודעתך בטיפול, נא להמתין!". כעבור מספר שניות הפניות הממתינות נקראות לזיכרון וזוכות לטיפול.

תכנית מיוחדת תטפל בתקלות השונות. אם זו תקלה בהקשה - תישלח אל המסוף הודעה מתאימה, אם התקלה היא בציוד התקשורת, תישלח הודעה אל מפעיל המחשב בתוך כדי ציון הקו או המסוף.

קיימות תכניות רבות לעיבוד נתונים בתקשורת ולא כאן המקום לפרטן. לסיכום, נציין עוד, שתכניות מיוחדות אלו לתקשורת מהוות מסגרת כללית לטיפול בתחומים השונים, שנזכרו לעיל. כל מפעל או ארגון, אשר מקימים רשת תקשורת, חייבים לספק לתכניות הללו את הפרמטרים המיוחדים, המאפיינים את הרשת ואת תכונותיה. לאחר מכן תיכתבנה התכניות המתאימות, אשר ככל תכנית יישום, תעבורנה את כל אותם השלבים, אשר תוארו בפרקים 5 עד 7.

11.7 עיבוד נתונים מבוזר

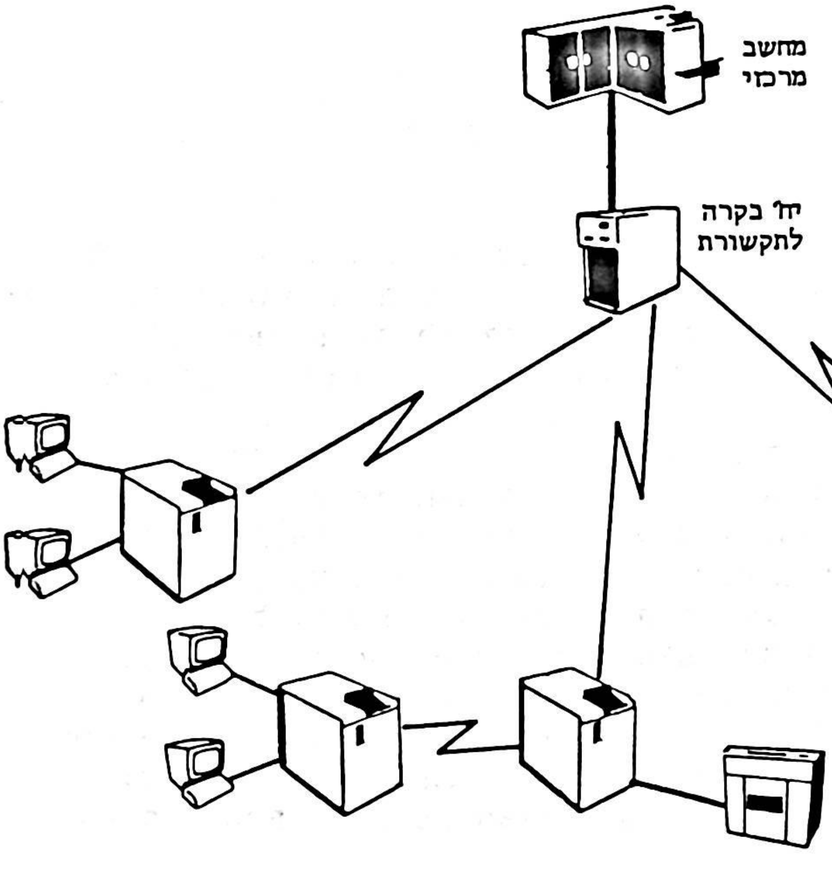
ההתפתחות הכלכלית המואצת, אשר איפיינה את שנות ה-60 וה-70 בעולם המערבי, הצמיחה ארגונים גדולים אשר השתרעו על פני ארצות, ולעתים אפילו יבשות אחדות. עבור הנהלות ארגונים כאלה, המחשב הפך להיות לאחד הכלים החשובים ביותר של ניהול ובקרה.

אולם, לקראת המחצית השניה של שנות ה-70 חלה תפנית מעניינת. טכניקות ניהול חדישות הטיפו לפיזור האחריות ולהאצלת סמכויות להנהלה המקומית המרוחקת. מנהלים אזוריים ביקשו אפשרות לפקח על המשאבים הארגוניים, שהועמדו לרשותם בצורה יעילה ועדכנית. כלי הניהול החשוב, המחשב, היה מרוחק והקמת רשת תקשורת היתה כרוכה בהוצאות עצומות. גם בתחום הטכנולוגיה של המחשבים חלה התקדמות ניכרת: הופיעו מיני-מחשבים זולים ומסופים מתוחכמים, שמחירם לא היה גבוה. שני התהליכים הללו סללו את הדרך לשיטת עיבוד מיוחדת: עיבוד נתונים מבוזר (DISTRIBUTED DATA PROCESSING).

נתאר לעצמנו ארגון גדול, בעל סניפים המרוחקים זה מזה מרחק ניכר. ארגון כזה יכול להיות בנק גדול או רשת סופרמרקטים. להנהלת הארגון יש מחשב מרכזי גדול, המשרת בעיקר את צרכיה. המחירים הזולים יחסית של ציוד לעיבוד נתונים, מאפשרים להציב בכל סניף מחשב קטן עם מספר מסופים וההנהלה אמצה שיטה זו. הבה נבחן כיצד היא פועלת.

לבנק גדול, בעל מאות סניפים ברחבי הארץ, יש מחשב מרכזי גדול (לעתים יותר ממחשב אחד) המשרת את ההנהלה המרכזית של הבנק, ומטפל במאגר הנתונים המרכזי, בעזרת ספרית תכניות ענקיות. בכל סניף ממוקם מחשב קטן, אליו מתוברים מסופים מסוגים שונים. חלק מהמסופים מופעל על ידי הפקידים, המשרתים את הקהל. מסופים מסוג אחר הינם מדפסות, שכל לקוח בעל כרטים אשראי של הבנק יכול להפעילן. גם המכשיר האוטומטי המנפיק שטרי כסף, לאחר הקשת המספר הסודי וכו', מחובר, כמובן, אל המחשב. המחשב המקומי הקטן מחובר אל המחשב המרכזי דרך קווי תקשורת. ראה ציור 11-22.

בתנאי תקשורת תקינים, כל פקיד בסניף כלשהו יכול להפעיל תכניות ישום מספריית התכניות המרכזית לשם שאילתא, עדכון וכדי. במקרה של תקלה בתקשורת, המחשב המקומי, אשר שימש עד כה כתחנת מעבר, "לוקח את הפיקוד". המחשב המקומי מאפשר לבצע את כל הפעולות הרגילות המקובלות, כגון משיכה והפקדה של כספים, חישוב עמלות, קבלת הוראות שונות לביצוע מאוחר יותר, הדפסת חשבונות ועוד. למחשב המקומי מאגר נתונים קטן, בו נרשמת כל פעולה. כאשר מתחדש הקשר עם המרכז, המחשב המרכזי קורא את כל הפעולות, שנרשמו במאגר הנתונים של הסניף ומעדכן את הקבצים המתאימים.



ציור 11-22 - מערכת "עיבוד נתונים מבוזר"

דוגמת הבנק הינה רק אחת מיני רבות בתחום של עיבוד נתונים מבוזר. דוגמה אחרת יכולה להיות זו של רשת חנויות סופרמרקט, או מפעל גדול, אשר גם מייצר וגם משווק את תוצרתו.

המחשבים הממוקמים באתרים השונים, משמשים בעיקר את עובדי האתר המסוים. אנשי המחסן מפעילים מחשב קטן, ובעזרת מסופיו הם מנהלים את המלאי בצורה יעילה. אנשי מרכז ההפצה משתמשים במחשב משלהם כדי לבצע את משלוח הסחורות בצורה יעילה ובמועד הנדרש. המחשב המרכזי של המפעל משרת את ההנהלה ואת מחלקות המטה. אולם, הקשר הקיים עם המחשבים המקומיים מאפשר שליפת מידע מדוייק ועדכני לצורך קבלת החלטות. נמחיש זאת בדוגמה.

מפעל מסוים מייצר פריט, אשר מפעל אחר משתמש בו כמרכיב חשוב למוצריו. במפעל ההרכבה פורצת שריפה ועל מנת לעמוד בהתחייבויותיו, הוא מוכן לשלם מחיר גבוה יותר עבור אספקה של 5,000 יחידות מהפריט הזה בתוך שבוע ימים.

מנהל המפעל זקוק למידע רב כדי להחליט, האם לספק את הסחורה. עליו לדעת מה מצב המלאי כרגע, מה מצב ההזמנות, מהן ההתחייבויות ללקוחות השונים, האם הוזמן מספיק חומר גלם לעתיד הקרוב, ועוד ועוד. רק לאחר קבלת <u>כל המידע מכל המקומות</u> האפשריים, הוא יגיע להחלטה האם הוא יכול להסכים לעסקה מפתה זו או לא.

לסיכום, הבה נראה מהם היתרונות והחסרונות של עיבוד נתונים מבוזר.

<u>יתרונות</u>

1. היחידה המקומית זוכה במחשב. הכלי חשוב לניהול תקין ולבקרה יעילה;

2. רוב הנתונים עוברים את תהליך העיבוד במקום בו נוצרו והפלט מופק במקום שנזקקים לו;

3. ניתן להשיג אחידות בתכנה המופעלת בסניפים השונים, מכיון שהיא מסופקת מן המרכז;

4. המסופים המחוברים למחשב המקומי מאפשרים עיבוד הידברותי, אשר מעלה את רמת השרות הניתן ע"י הארגון;

5. עיבוד הנתונים השוטף אינו תלוי בסדר העדיפויות של המחשב המרכזי, שנקבע לרוב <u>לא</u> לפי הצרכים של הסניפים;

6. בעיבוד נתונים מבוזר כל סניף נהנה מקצב עבודה משלו, אשר אינו תלוי בגורם חיצוני אחר.

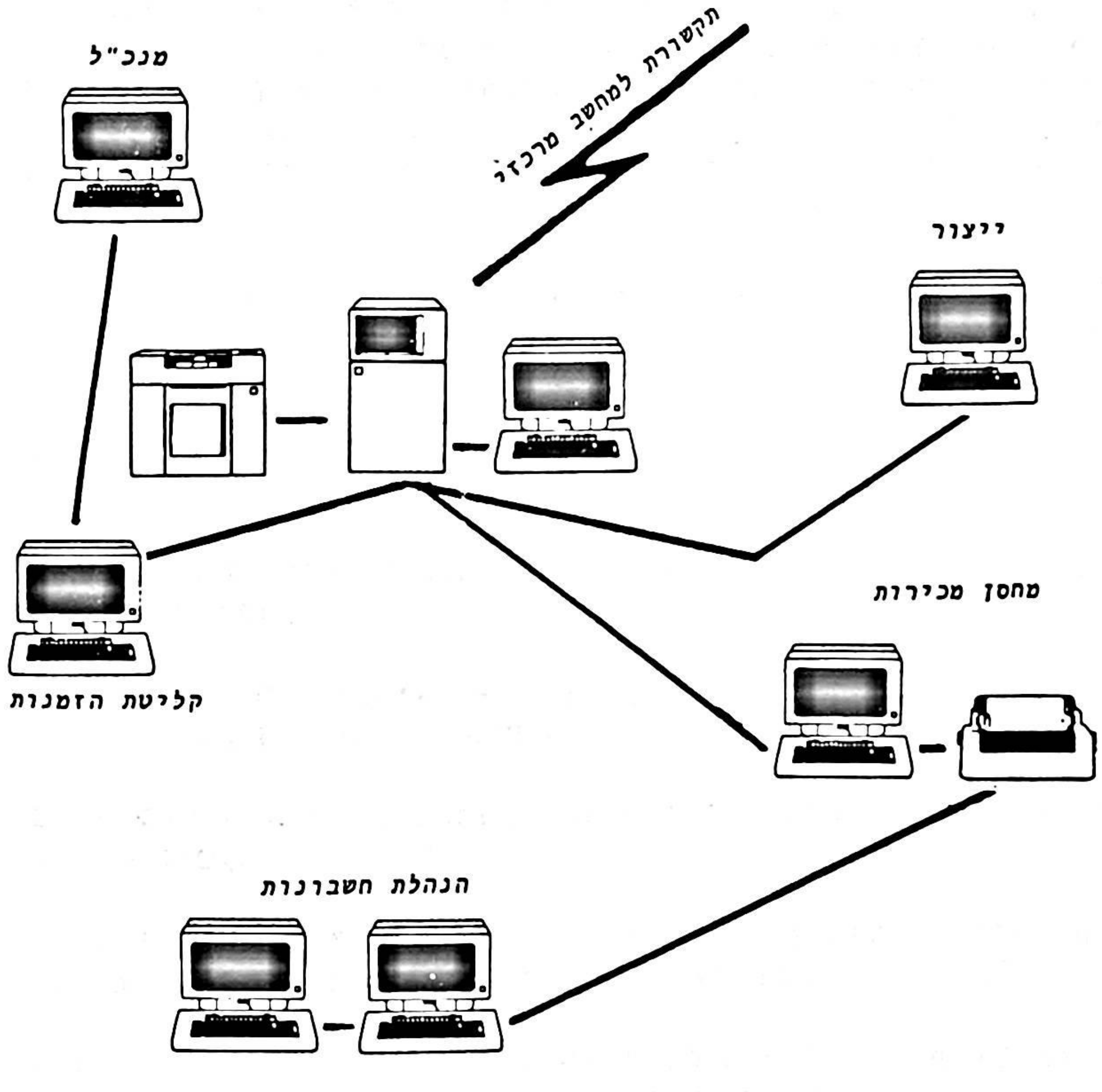
חסרונות

1. עיבוד נתונים מבוזר יכול לפגום באחידות של נוהלי ושיטות העבודה בארגון כולו אם מפתחים תוכנות שונות;

2. עיבודים מורכבים חייבים להעשות במחשב המרכזי הגדול וזה יכול לגרום לצווארי בקבוק;

3. המחשב המרכזי חייב לספק כח-אדם לפיתוח והחזקת התכנה גם עבור הסניפים או להכין עבורם תכנה;

הנטיה היום היא לשילוב של מחשבים מקומיים ומחשב מרכזי, אשר ידגיש את יתרונות השיטה וימעיט, ככל האפשר, את החסרונות.



פרק 11 - שאלות

1. מה ההבדל בין עיבוד נתונים באצווה לבין עיבוד נתונים בתקשורת?

2. "זמן תגובה" מהו? האם חשוב למדוד אותו בכל יישום ויישום?

3. מנה את המרכיבים של רשת תקשורת.

4. לאיזה סוג של יישום מתאים לדעתך קו חד-מגמי?

5. מהם החסרונות בשימוש בקו רב-נקודתי?

6. הבחן בין חקירה לבין מעינה.

7. מהם ההבדלים בין השיטה הסינכרונית לבין השיטה האסינכרונית?

8. אנשים מסוימים משווים את התכנה לע"נ בתקשורת למשגות, אשר הוזכר בפרק 8. האם יש הצדקה לכך?

9. "עיבוד נתונים מבוזר יגרום, בסופו של דבר, לחיסול המחשבים הגדולים". הגב על הציטטה.



״יש לי יחס סנטימנטלי למַקש מיוחד זה.
הוא עושה כעת את העבודה אשר אני עשיתי
כאשר התחלתי את עבודתי כאן״.

פרק 12 – מיני, מיקרו והמחשב האישי

כל אחד מאתנו השתמש, או לפחות ראה בעבר מכונות חישוב מכניות, אשר במשך עשרות בשנים ביצעו את פעולות החישוב השונות במהירות ובתוך כדי השמעת רעשים.

עד לפני מספר שנים היה ההבדל בין מכונת חישוב לבין המחשב ברור: מחשבים שכנו במרתפים או בחדרים בעלי רצפה מיוחדת, ואליהם התחברו קוראות כרטיסים, סרטים מגנטיים ומדפסות גדולות, בעוד שמכונות החישוב נמצאו במשרדים או בבנקים, ו"התהדרו" בכפתורים ובמתגים לעשרות.

כיום ההבדל אינו כה ברור. גם מכונות חישוב מתחברות אל מדפסות זעירות, סרטים ומסכי טלויזיה. היכן נמצא הגבול המפריד בין מכונת החישוב והמחשב?

12.1 מכונות חישוב אלקטרוניות

מכונות חישוב אלקטרוניות הופיעו באמצע שנות ה-60. הן ה מהירות ושקטות, כי פעלו בעזרת סוללות, אך מחירן היה גבוה – כ-2,000 דולר! הודות לשיפורים טכנולוגיים מתמידים והודות לשימוש יעיל במעגלים מודפסים הלך וקטן גודל מכונת החישוב, ויחד עם זאת הלך וירד המחיר. בשנת 1984 ניתן היה לקנות מכונת חישוב (הנקראת בפי העם "מחשב-כיס") בפחות מ-10 דולר!

בקבוצה של מכונות החישוב ניתן להבחין במספר סוגים:

1. מכונה סטנדרטית, בעלת ארבע פעולות חשבון.

2. מכונה בעלת אפשרות פעולות מורכבות.

3. מכונה שניתן לתכנת אותה בצעדי תכנות פשוטים, בשפת מכונה.

4. מכונת חישוב שניתן לתכנת אותה עם תכנית בשפה עילית, וניתן לחבר אליה קלטת ומדפסת קטנה.

המשוכללות והיקרות שבין מכונות החישוב הללו מכילות גם תכניות למשחקים שונים, ומבינות שפות כמו BASIC.

אולם לכל סוגי המכונות הללו יש כמה תכונות משותפות, אשר מאפיינות אותן כ"מחשבי-כיס" - **<u>ומבדילות</u>** אותן ממחשבים:

1. הנתונים או "חומר הגלם" אותו הם מסוגלים לעבד - הינו מספרים בלבד.

2. הכנסת הנתונים נעשית בצורה איטית, על המקלדת של המכונה.

3. הפלט מופיע בצורה מספרית, על-פי-רב ללא אותיות.

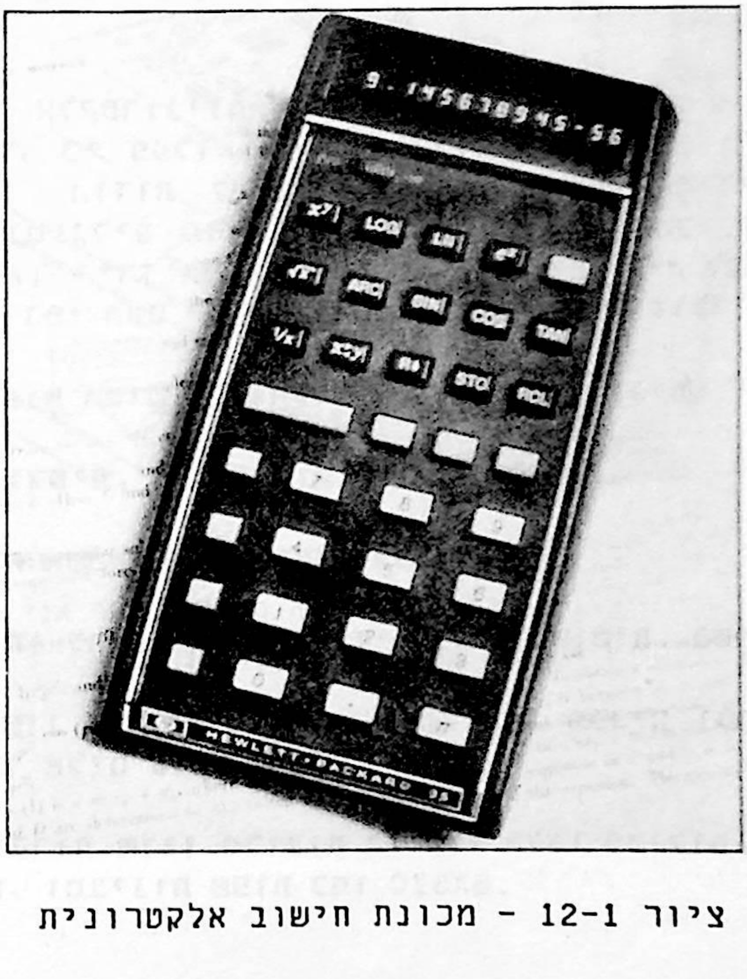
4. מכונות אלו דורשות פעילות ונוכחות מתמדת מצד האדם המפעיל אותן.

5. האדם "עובד" זמן ממושך יותר מאשר המכונה.

6. מכונות החישוב חסרות מעגלי בקרה משוכללים, ולכן כל פעולה מתבצעת אך ורק כאשר לוחצים על הכפתור המתאים, וכל תקלה מחייבת לחזור על שלבי עבודה לפי שיקול המשתמש.

7. גודל הזכרון מוגבל, ולכן גם גודל התכנית, שניתן לאחסן בו מוגבל. לרב - מספר עשרות צעדי עיבוד.

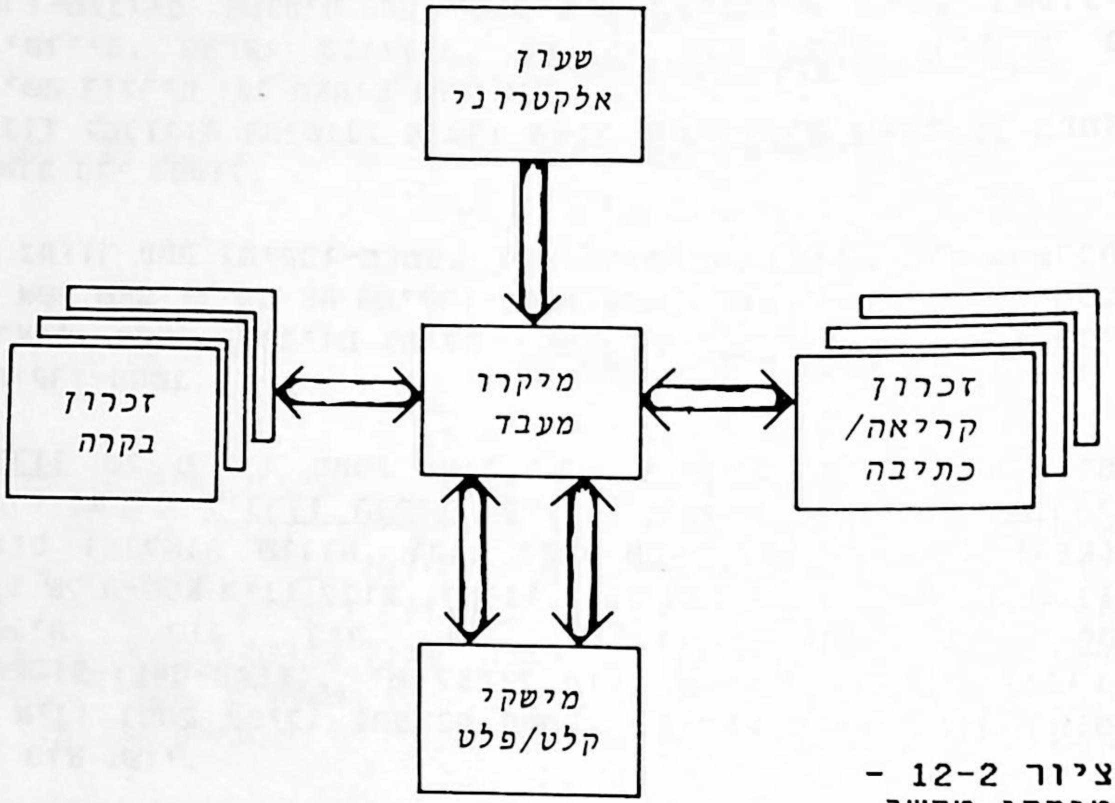
8. הנתונים מוזנים בצורה עשרונית.



ציור 12-1 - מכונת חישוב אלקטרונית

המחשב הקטן ביותר נקרא **מיקרו-מחשב** (MICRO-COMPUTER), או מחשב זעיר. הוא לא רק קטן בגודלו, אלא מכיל פחות חלקים מאשר מחשבים בינוניים וגדולים וגם פעילותו מוגבלת יותר. הבה נתאר אותו:

מיקרו-מחשב הינו מכונה המורכבת מחמישה חלקים עיקריים, כפי שמורכבים גם "אחיו" הגדולים. ראה ציור 12-2.



ציור 12-2 - מיקרו-מחשב

בלב המערכת נמצא **המיקרו-מעבד** (MICRO-PROCESSOR). ייצור המיקרו-מעבדים קשור בשמו של **טד הוף** (HOFF), אשר הצטרף בשנת 1969 לחברה בשם "אינטל", שנוסדה באותה עת. הוף נתבקש לתכנן מרכיבים זעירים למכונות חישוב קטנות ("מחשבי-כיס"). לאחר עבודה של שנה בערך הצליחו טד הוף וחבריו לרכז 2250 טרנזיסטורים של יע"מ על שבב סיליקון, שגודלו היה כגודל ציפורן האדם. כוח החישוב של המעבד החדש השתווה לזה של אחד המחשבים הראשונים (ENIAC,1946), שמחירו היה בזמנו מיליון דולר! מעבד קטן זה היה מסוגל לבצע את הפעולות הבאות:

1. קבלת נתונים בצורה בינרית (0, 1).

2. שמירה על הנתונים למועד עיבוד מאוחר יותר.

3. ביצוע פעולות אריתמטיות ולוגיות, בעזרת הוראות מתאימות שאוחסנו בזכרון קודם לכן.

4. העברה של תוצאות העיבוד ליחידת פלט כלשהי.

מיקרו-מעבדים מסווגים בד"כ על פי גודל המילה, אשר נמדדת במספר סיביות:

4 BIT	16 BIT
8 BIT	32 BIT

מספרים אלה משקפים, למעשה, את גודל <u>האוגרים</u> (REGISTERS), בהם נעשית פעילות החישוב.

מיקרו-מעבדים פשוטים הם המרכיבים העיקריים היום בשעונים דיגיטליים, משחקי טלויזיה, משאבות דלק וכוי, ומחירם נע מחמישה דולרים ועד מאתים וחמישים דולר.
בניגוד למכונות החישוב, מיקרו מעבד <u>איננו דורש</u> התערבות מתמדת של אדם כדי לפעול.

הבה נחזור עתה למיקרו-מחשב. המיקרו-מעבד, הנמצא בלב המערכת, הוא אשר מאפיין גם את המיקרו-מחשב עצמו. מהירות הביצוע, מספר ההוראות, מספר הסיביות במילה - אלה הם הקריטריונים העיקריים של מיקרו-מחשב.

<u>הזיכרון</u> של מיקרו מחשב עשוי גם הוא משבבי סיליקון, בדומה למיקרו-מעבד. <u>זיכרון הבקרה</u> מכיל את מערכת ההוראות, נתונים קבועים וטבלאות שונות, והוא נקרא <u>ROM</u> (READ-ONLY-MEMORY). תוכנו של ה-ROM הינו קבוע, ואיננו ניתן לשינוי ע"י פעולת תכנות חיצונית. סוג כזה של זיכרון נקרא גם <u>ROS</u> (READ-ONLY-STORAGE). יש לציין עוד, שהתוכן של תאי זיכרון אלה איננו נמחק אפילו בהפסקת חשמל, כי הוא צרוב בתוך החומר ממנו הוא עשוי.

סוג זיכרון אחר נקרא <u>RAM</u> (RANDOM ACCESS MEMORY), אשר משמש כ"שטח עבודה". ניתן לכתוב בו ולקרוא ממנו. זיכרון זה גדול על-פי-רב יותר מאשר ה-ROM. שני הבדלים עיקריים קיימים בין סוג זה של זיכרון לבין הסוג הקודם:

1. תוכן ה-RAM איננו קבוע. הוא יכול להשתנות ע"י הוראת תכנות, לפעמים מכילים תאי ה-RAM נתונים שונים, ולעתים הם מכילים את התכנית, אשר מעבדת נתונים.

2. תוכן ה-RAM נמחק עם כיבוי המכשיר, או אם יש הפסקה בזרם החשמל, המסופק למכשיר.

<u>12.3 מיני מחשבים</u>

לא רב המרחק בין המיקרו-מחשב לבין <u>המיני-מחשב</u> (MINI-COMPUTER), או <u>מחשב קטן</u>. אפילו בעלי מקצוע ותיקים

ומעולים בשטח המחשבים מתקשים להגדיר מה הוא מיני-מחשב. כיום ישנה הסכמה, שמיני-מחשב נמצא בין המיקרו-מחשבים והמחשבים הגדולים והבינוניים. הוא דומה יותר למחשבים הגדולים, כי הוא אינו זעיר כמו מיקרו-מחשב, ואינו מוגבל בפעולותיו ובזיכרונו. בנוסף לכך ניתן לחבר אליו יחידות קלט/פלט מסוגים רבים, כמו אל המחשבים הגדולים. אולם בכל זאת קיימים מספר הבדלים בין המיני מחשבים ובין "אחיו" הגדולים.

- המיני זול יותר.
- המיני קטן יותר, יש לו פחות אוגרים, "פחות" זיכרון.
- מהירות הביצוע נופלת בהרבה מזו של ה"גדולים".
- יחידות קלט/פלט, אשר מתחברות אל המיני הן איטיות ופשוטות יותר.

12.4 שימושים של מיקרו-מחשבים

בתחילה הופיעו מיקרו-מחשבים בתעשיה. תפקידם היה להבטיח את האוטומטיזציה של פעילויות שונות או לשמש כמכשירי בקרה מיוחדים ומדויקים. מיקרו-מחשבים מילאו תפקידים כאלה בתעשיות המטוסים והאוניות, הם הופיעו במעבדות כימיות ורפואיות, מדדו עוצמות של לחץ וגובה של טמפרטורה. בגלל גודלם המוגבל ניתן היה להכניסם למקומות קטנים וצרים. הם משמשים כמרכיב עיקרי בתעשיית רובוטים, טילים מונחים ומכשירים רפואיים שונים.

מאוחר יותר הופיעו מיקרו-מחשבים גם במשרדיהם של עורכי-דין, רופאי שיניים וחברות ביטוח, ואיפשרו לבצע פעולות שונות בתחום הנהלת החשבונות. בכל הפעילויות הללו באו לידי ביטוי אותן תכונות אופייניות למחשבים, אשר מבדילות אותם הבדלה ברורה וחד-משמעית ממכונות החישוב, אשר הוזכרו בתחילת פרק זה:

1. מחשבים עובדים <u>גם</u> עם <u>נתונים אלפא-נומריים</u> בנוסף לנתונים המספריים. שמות, כתובות, חלקים במחסן, טקסטים ועוד - כל אלה משמשים גם כן כחומר גלם למחשב. ניתן למיין שמות לפי אל"ף-בי"ת, ניתן לערוך מכתבים או טקסטים, להעתיק אותם, לתייק אותם וכו'.

2. ניתן לאחסן בזיכרון <u>תכניות</u> מחשב, אשר <u>יעבדו ללא נוכחות האדם,</u> באופן אוטומטי, לאורך זמן.

3. <u>התכניות ניתנות להחלפה או לשינוי.</u> מספר התכניות, שמחשב יכול לקלוט בלתי מוגבל להלכה, והדבר היחיד המגביל כאן הינו גודל הזיכרון ואמצעי האחסון שלו.

4. כתוצאה מן העובדה, שניתן לחבר <u>יחידות קלט/פלט</u> שונות, קליטת הנתונים מבוצעת במהירות רבה מאד. הפלט מוצג למשתמש במגוון רחב של צורות, אשר תלוי רק ביחידות הפלט, המתחברות אל המחשב.

5. במחשבים הללו מורכבים <u>מעגלי בקרה</u> משוכללים, ולכן ניתן כאן לבצע תכניות ארוכות, ללא השגחה צמודה של בני-אדם. בעת תקלה בביצוע יכול המחשב לפתור בעצמו חלק, אם לא את כל הבעיות, ולהמשיך בביצוע.

12.5 למה מחשב אישי?

השנים הראשונות של העשור הנוכחי מתאפיינות, בין היתר, גם בחדירת המחשב לתוך הבית. בתחילה היו אלה משחקי טלויזיה, אך אט-אט נדחקו רגליהם של משחקים אלה ע"י <u>המחשב האישי</u> (PERSONAL COMPUTER).

המחשב האישי "התחיל את דרכו" כמחשב זעיר, והצטיין בשתי תכונות חשובות: מחיר נמוך ונוחות הפעלה. היצרנים הוציאו לשוק, זו הפעם הראשונה, מחשב שהוא יחידה שלימה, אשר כוללת יחידת עיבוד, מסך ומקלדת. המחשב מחקה את "אחיו" הגדולים יותר אך הוא זול מהם במחירו ואיטי מהם ומוגבל בביצועיו. עצמתו אינה בחמרה אלא בתכנה. התכנה ה"ידידותית" והיכולת של כמעט כל אחד לקבל את שרותיה המגוונים הם סוד הצלחתו של המחשב האישי. תכנה זו הינה רב-גוונית עד כדי כך, שהיא מאפשרת שימוש במחשב האישי הן במשרד והן בבית. במשרד ניתן להשתמש בו, הודות לתוכנות מיוחדות להנהלת חשבונות, לניהול מלאי, לטיפול בפוליסות של חברות ביטוח או להחזקת כרטסת של רופא שיניים - והרשימה עוד ארוכה. ובבית?

ובכן, בבית יכול המחשב האישי לשרת כל אחד מבני המשפחה. נתחיל בילדים. הם יכולים לשחק "איתו" בכל משחקי טלויזיה, שנמכרים היום לעשרות ואולי למאות בכל רחבי תבל. אך לא במשחקים המדובר. תכניות לימוד מיוחדות מופצות היום כמעט בכל המקצועות: חשבון, היסטוריה, דקדוק, תורת המוסיקה ואפילו מדעי המחשב! התכניות נמכרות על סרטים, קלטות או דיסקטים, ומותאמות כמעט לכל הסוגים של המחשבים האישיים.

המשפחה יכולה להיעזר במחשב האישי בניהול תקציב המטבח, בהחזקת קובץ התפריטים וככלי לגילוי שריפה, אמצעי אזעקה או אפילו כשמרטף (עם חיבור מתאים אל השכנים, כמובן...).

אבי המשפחה יכול להשתמש במחשב האישי ככלי לניהול הוצאות המשפחה, התכתבויות, ספרייה פרטית, לימוד שפות זרות; ואם קיים האביזר המתאים, גם לעיבודים גרפיים. בארה"ב ניתן כיום

להתחבר, בעזרת ציוד מיוחד, אל רשת תקשורת, והמחשב האישי יכול לשלוף מידע ממרכז עירוני, מאוניברסיטה או מספרייה מקומית. שרות זה מוכר בשם <u>וידיאוטקס</u> (VIDEOTEX).



ציור 3-12 - מחשב אישי PC/AT של יבמ

מחשב אישי נפוץ מאד הוא PC של חברת יבמ, הוא כולל מיקרו-מעבד של 16 סיביות, גודל הזיכרון נע בין 64K לבין 544K (RAM). ניתן לרכוש אותו עם כונני תקליטונים, ו/או עם כונני דיסקים קשיחים אשר מרחיבים את הזיכרון החיצוני עד 20MB. ניתן לחבר אליו מגוון רחב מאד של מדפסות. למחשב חדש בסדרה תכונות חדשות: גודל הזיכרון שלו מגיע ל-3MB ותכולת הדיסקים ל-40MB.

ספריית התכניות כוללת מערכת הפעלה הנקראת DOS. ניתן לתכנת אותו בשפות רבות: BASIC, APL, PASCAL, COBOL ואסמבלר. המחשב "יודע" עברית, הוא "מצייר" ו-"מנגן", ומה לא...?

12.6 דוגמאות ליישומים עם מיקרו-מחשבים

נסיים את הדיון הקצר בהצגת שני יישומים עם מיקרו-מחשבים: האחד - בתחום הבקרה התעשייתית, והשני - בתחום המשרד.

1. כיצד אפשר לחסוך בחשמל?

ההוצאות על שימוש בחשמל הולכות וגדלות, ככל שעולה מחיר הדלק, שבעזרתו מפיקים חשמל. מובן מאליו, שמאמצים רבים מושקעים כל העת במציאת פתרון נאות לבעיה כאובה זו.

אחד הפתרונות קשור בענף המחשבים, וייחודו של פתרון זה, בכך שהוא פותח בישראל ע"י חברת מוטורולה בארץ. הפתרון הזה אינו מוזיל אמנם את מחירי הדלק או החשמל, אבל הוא מפקח על השימוש בחשמל, ובכך מאפשר חסכון כספי של קרוב ל-עשרים וחמישה אחוז!

מערכת מחשב זו, ששמה MEIC-3000, מורכבת משני חלקים עיקריים:

- מרכז הפעלה הכולל מחשב קטן, מקלדת, יחידת מסך וכן יחידה מיוחדת הקשורה למצברים, למקרה של הפסקת חשמל.

- יחידות-קצה, המותקנות סמוך למתקנים המבוקרים. ניתן לחבר עד כ-250 מתקנים נפרדים.

מערכת זו מבטיחה שליטה ובקרה על צריכת האנרגייה, והיא מאפשרת לווסת עומסים של מיזוג-אויר, תאורה ואוורור בצורה מחזורית. הפיקוח המרכזי בעזרת המחשב מונע גידול בשיא הביקוש, ומשנה משטרי עבודה של המערכת בהתאם למידות החום. כמו כן ניתן לצבור נתונים לגבי העומסים, לרשום תקלות, להדפיס דו"חות יומיים וכדי.

תוכנה חשובה של המערכת MEIC-3000 היא זו המטפלת בשיא הביקוש. זוהי תכנית, שמטרתה לצמצם את עלות צריכת החשמל בגלל צריכה גבוהה במיוחד של אנרגייה בפרקי זמן קצרים. התכנית עוקבת אחרי מגמות הצריכה במקומות של העומסים השונים, ומנתקת באופן הדרגתי מרכזי צריכה שונים סמוך לחצייה של נקודת שיא הביקוש.

2. עיבוד תמלילים (WORD PROCESSING)

בפרק הראשון של הספר הזה תואר התהליך של <u>עיבוד נתונים</u>. צוין שם, שתהליך זה הופך את <u>הנתונים</u> (חומר הגלם) <u>למידע</u>

(מוצר מוגמר). המונח **עיבוד תמלילים** (WORD PROCESSING) הוטבע רק בשנים האחרונות, והוא מדגיש את הייחוד של חומר הגלם, אשר מעובד כאן. חומר הגלם המורכב ממשפטים, סעיפים, מכתבים, דו"חות, בקיצור - טקסט, או תמליל.

עיבוד תמלילים מציין כיום שיטת עבודה חדשה, אשר חדרה לתוך המשרד המודרני, וכבשה אותו בסיוע לא מבוטל של המיקרו והמיני מחשבים.

הבה נשחזר את הפעילויות של עובדי משרד כלשהו, בשיטת העבודה המסורתית:

מזכירה או כתבנית מקבלת להדפסה מסמך הכתוב בכתב יד. היא מדפיסה אותו כטיוטה ראשונה, ומחזירה אותו למחבר. המחבר עובר על המסמך, מתקן את הטעון תיקון, עורך שינויים קלים, ומחזיר אותו למשרד.

ברב המקרים חייבת הכתבנית להדפיס מחדש את כל המסמך. לעתים עובר המסמך תיקון והגהה נוספים, ומודפס פעם שלישית ורביעית. בתוך כדי העבודה הזאת עסוקה הכתבנית, כמובן, גם בדברים אחרים, כגון: שיחות טלפון, טיפול ביומן הפגישות של המנהל, תיוק וכו'. לבסוף מוכן המסמך, והכתבנית ניגשת אל מכונת הצילום ומצלמת כל דף במספר העותקים הדרוש. לאחר מכן אוספים את הדפים למסמכים שלימים, לשם הפצה. מכינים מעטפה עבור כל מכותב, מכניסים בה את העותק המיועד לו, ושולחים בדואר. הזמן העובר בין חיבור המסמך לבין שליחתו בדואר - כיומיים בממוצע.

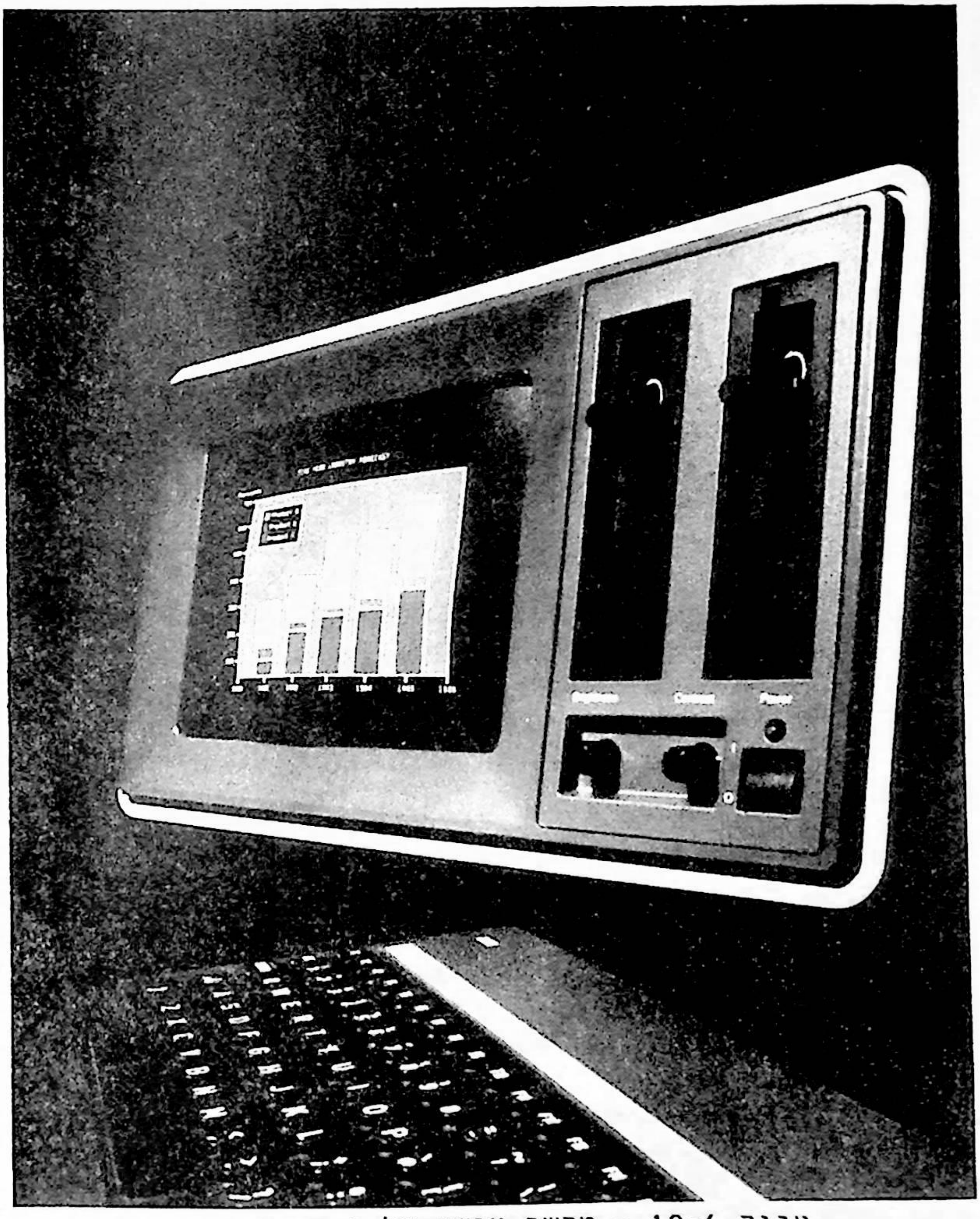
השיטה של עיבוד התמלילים מציעה מהפכה בחיי המשרד המודרני. בעזרת מערכת מחשב קטנה (לרב מיקרו או מחשב אישי) ניתן לחסוך בזמן, בכוח-אדם ובמשאבים אחרים. שיטת עבודה זו חוסכת גם כמות רבה של נייר, בנוסף ליתרונותיה הרבים האחרים. יעילות השיטה באה לידי ביטוי בשלבים השונים של עבודת המשרד:

א. **הדפסת מסמך**. המסמך "מודפס" בתחילה על מסך של מצג. אם הכתבנית טועה בכתיב או מדלגת על מלה או שורה, קל מאד לתקן זאת. בלחיצת מקש ניתן להזיז אותיות, מלים או שורות שלימות. בגמר יצירת המסמך הוא מאוחסן בדיסק או בתקליטון, וניתן לשליפה בקלות.

ב. **עריכה**. ניתן לערוך את המסמך בצורות מגוונות, אפשר לקבוע את אורך השורה (40 או 100 תווים לדוגמה), אפשר להזיז קטעים שלימים בגוף המסמך או לצרף מספר קטעים. עימוד ומספור הם אוטומטיים. ניתן ליישר שוליים גם בצד הימני וגם בצד השמאלי של המסמך, ועוד.

ג. __הפצה.__ לאחר השלמת המסמך הוא נשלח למכותבים בדפוס (או ברשת תקשורת, אל המחשבים שלהם). כל מכותב יכול לקבל מספר עותקים, כפי דרישתו.

ד. __תיוק ושליפה.__ כשמסיימים את הכנת המסמך, "מתייקים" אותו באמצעי אחסון מגנטיים בשיטה אשר מאפשרת את איתורו במהירות רבה. איתור המסמך יכול להיעשות בכמה דרכים: לפי שמו, תאריך הדפסתו או מילת מפתח כלשהי בגוף החומר. כאשר המסמך עובר שינויים, ניתן לתייק את המסמך החדש, ולהשאיר גם את המקור בקובץ.



ציור 4-12 - מחשב אישי של נ.סי.ד.

פרק 12 - שאלות

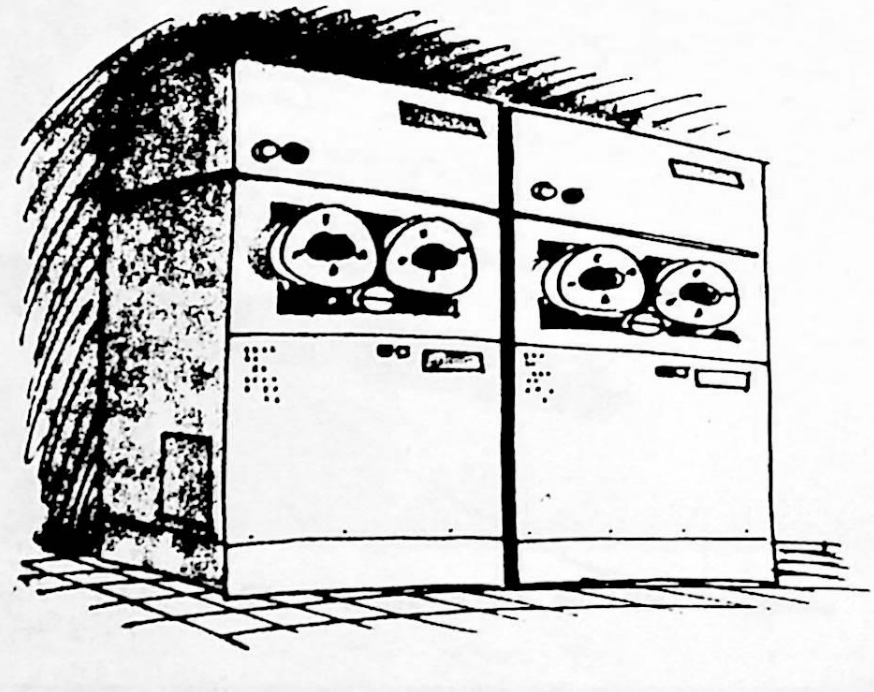
1. מדוע אין לראות במכונת חישוב אלקטרונית - מחשב?

2. מהו תפקיד האוגר? האם יש קשר בין האוגר לבין גודל המלה במיקרו-מעבדים?

3. "היום אין הבדל בין מיקרו-מחשב לבין מיני-מחשב", התייחס אל טענה זו.

4. "המחשב האישי הופך להיות לכלי שימושי המוני, כמו הטלפון, הרדיו, והטלויזיה". האם נכונה טענה זו?

5. הסבר בקצרה את המונח "עיבוד תמלילים".



"זכור את דברי - לא ירחק היום
והמיקרומחשבים יתפשו את מקומנו"



ציור 5-12 - מחשב אישי של HP

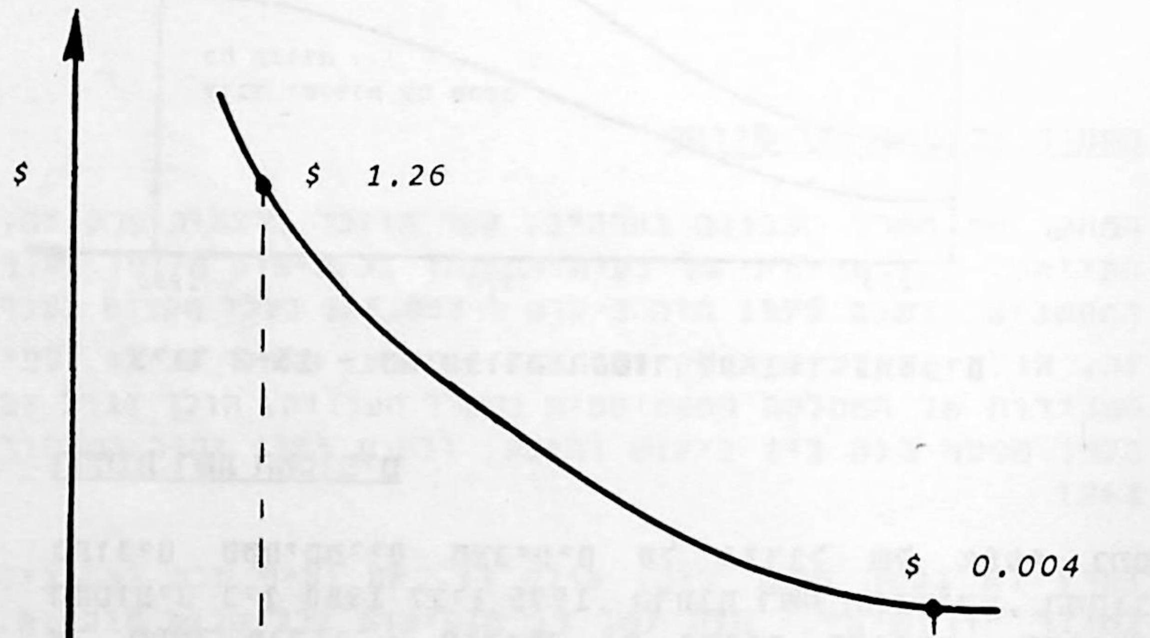
פרק 13 - מרכל מידע

קצב ההתפתחות המואץ בכל הקשור בעיבוד נתונים ממוכן הביא לכמה תופעות, שלא נחזו מראש. התפתחות זו מצאה את ביטויה בשטחים שונים של חיינו: במשרד, בבית החרושת, בבית הספר ואפילו בבית. הבה נבחן תופעות אלו, אשר משמשות למעשה רקע כללי למצב הולך ומחמיר, אשר דורש פתרון בעתיד הקרוב ביותר.

13.1 ר ק ע

1. התפתחות טכנולוגית

במשך שנות ה-70 של המאה שלנו פותחו שיטות ייצור חדישות שבעזרתן ניתן היה לייצר שבבים של מחשבים בצפיפות גדולה מאד. לדוגמה - בקובייה קטנה, שנפחה אינו עולה על 0.5 סמ"ק, אפשר לדחוס כ-1 מליון סיביות! הודות לכך הולך וקטן גודלם הפיסי של המחשבים, ובעת ובעונה אחת יורד גם מחירם. בציור 1-13 מוצגת הירידה הדרסטית בעלות העיבוד במחשבים מאז 1953. אותה יחידת עבודה, שעלתה ב-1953 יותר מדולר ורבע, עולה היום 0.4 סנט!



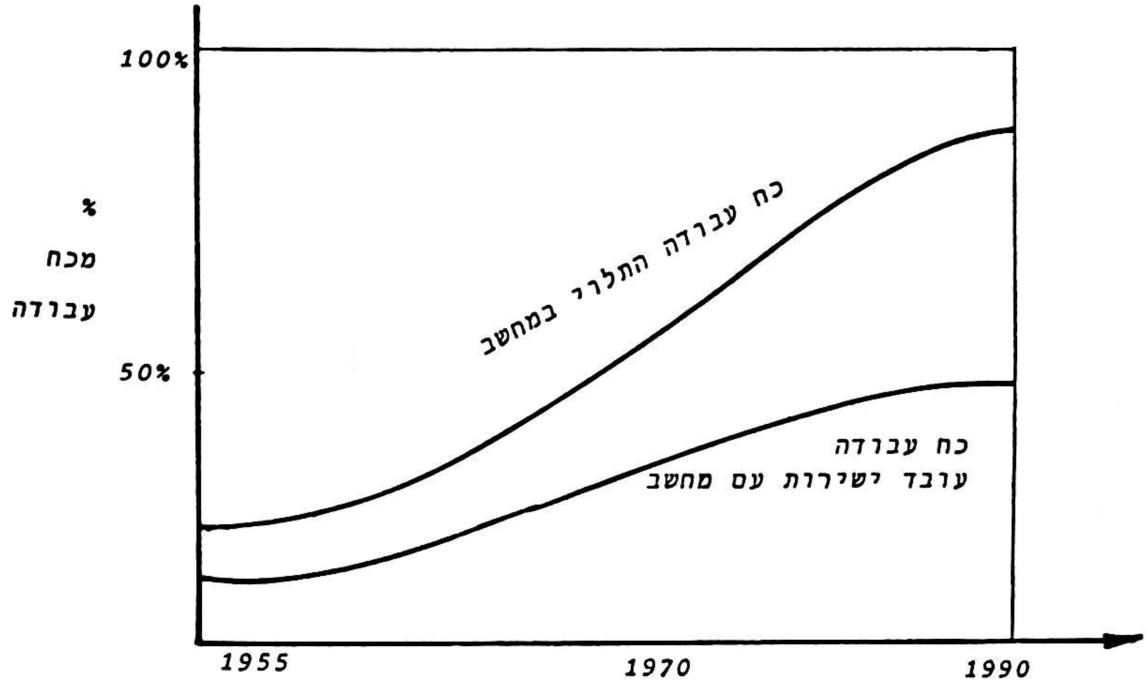
ציור 1-13 - הירידה בעלויות בענ"א

מחיר נמוך של ציוד ועלות נמוכה של עיבוד, גרמו לכך, שיותר ויותר מפעלים יכולים להרשות לעצמם לקנות או לשכור מערכת מחשב.

2. ציקת העובד אל עיבוד הנתונים

תופעה שניה, אשר למעשה נובעת באופן ישיר מן התופעה הקודמת, היא הקשר של העובד אל תהליך עיבוד הנתונים. ככל שימצאו יותר מחשבים בארץ, כך יהיו יותר עובדים מעורבים בשלב זה או אחר של תהליך עיבוד הנתונים.

בארה"ב נערך מחקר (משרד העבודה, ארה"ב 1980), שבעזרתו ניסו לחזות, איזה חלק מכוח העבודה יהיה קשור לעיבוד הנתונים. תוצאות המחקר, שמוצגות בציור 2-13 - מדהימות! מתברר, שבשנת 1990 ישתמשו כ-40% מכל העובדים בשרותי מחשב באופן ישיר, בעוד שכ-80% יהיו בקשר הדוק בצורה זו או אחרת אל עיבוד הנתונים.



ציור 2-13 - כח עבודה הקשור לעיבוד נתונים

3. הרחבת רשת המסופים

נתונים סטטיסטיים מצביעים על גידול של 400% במספר המסופים בין 1980 לבין 1995. הרחבת רשת התקשורת, המחוברת אל מחשב מרכזי - פירושה גם כתיבת תכניות, המתאימות לעבודה עם מסופים. תכנה זו מאפשרת עבודה הידברותית (אינטראקטיבית) לכל אדם היודע להפעיל מסוף. גידול במספר המסופים בקצב כזה רומז על אפשרות, שבסוף העשור הזה יהיה מסוף בכל מקום עבודה שני.

4. "המשתמש המתוחכם"

היום, יותר ויותר אנשים לומדים בקורסים הקשורים במחשבים. כמעט כל הסטודנטים באוניברסיטה חייבים לקחת לפחות קורס אחד של מחשבים; תלמידי תיכון יכולים לבחור היום מגמת מחשבים לבגרות. ואם המדובר דווקא בתלמידי בית הספר היסודי, משרד החינוך פרסם בשנה"ל תשמ"ג, כי ל-150,000 תלמידים בבתי הספר היסודיים איפשרו גישה אל מסופי מחשב באמצעות בית הספר. פירוש הדבר, שיותר ויותר אנשים נכנסים מידי שנה לקבוצת האוכלוסייה, היודעת להפעיל מסוף של מחשב ללא קושי. אך לפני חמש עשרה שנה היה חוג האנשים האלה מצומצם מאד בארץ, ומספרם לא עלה על אלפים בודדים.

5. ציפיות המשתמשים

לקראת סוף העשור הנוכחי יצפו המשתמשים במקום עבודה מסודר, לפחות לשני דברים:

- שיהיו די מסופים, שאותם ניתן יהיה להפעיל לצורך ביצוע העבודה השוטפת.
- שהמסופים ישמשו ככלי עבודה, אשר בדומה לטלפון, יהיו בעלי שתי תכונות חשובות:
  - פשוטים להפעלה,
  - עומדים לרשות המשתמש 24 שעות ביממה.

6. מחסור בכוח עבודה מיומן

המחקר של משרד העבודה בארה"ב, אשר הוזכר בראשית פרק זה, התייחס, בין היתר, אל בעית המחסור בכוח-אדם מיומן בענף המחשבים. בשנת 1975 היה ביקוש ל-320,000 בעלי מקצוע בענף זה, אולם ההיצע היה רק 308,000 - משמע, מחסור של 4%. לפי המדידות של המחלקה הסטטיסטית במשרד העבודה, הולך וגדל עם הזמן הפער הזה בין ביקוש והיצע, ובשנת 1985 יהיה המחסור 26%!

בארץ לא נעשה מחקר כזה, אולם די, אם נעיף מבט בעתונים במדור "דרושים". מזה זמן רב מופיעות בכל שבוע מודעות, המציעות עבודה לתכניתנים, ובמקרים רבים חוזרות אותן המודעות ומופיעות פעמיים ושלוש פעמים בחודש אחד.

את התמונה העגומה, שתוארה לעיל משלימה תופעה נוספת:

לחצים, המופעלים על יחידת ענ"א בארגון. הלחצים באים משלושה מקורות שונים:

א. <u>לחצים מצד המשתמשים:</u> המשתמשים טוענים, כי הם מחכים חודש ימים ויותר לדו"ח קטן, וכאשר מגיע הדו"ח לידיהם, אין הוא עונה על כל דרישותיהם. במקרים אחרים נתקלים המשתמשים בסירוב מוחלט לדרישה צנועה, לביצוע שינוי "קטנטן" בדו"ח או בקובץ. בנוסף לכך, פעמים רבות אין המשתמש מוצא שפה משותפת עם אנשי יחידת ענ"א.

ב. <u>לחצים מצד ההנהלה</u>

ההנהלה מתלוננת, כי כספים רבים הושקעו במערכת המחשב ופיתוח יישומים, ולמרות זאת קצב העבודה איטי, ההוצאות הולכות וגדלות וכל העת נשמעות תלונות מצד המשתמשים.

ג. <u>לחצים בתוך יחידת ענ"א</u>

רוב התכניתנים נאלצים לעסוק באחזקת תכניות קיימות, ורק פחות ממחציתם עוסק בעבודה בעלת אתגר מקצועי - קרי, פיתוח תכניות חדשות.

בגלל המחסור בכוח עבודה מיומן קיימת גם תחלופה גדולה של עובדים. יש לעובדה זו השלכה על יעילות התפעול של היחידה, אפשרות התכנון, ועוד.

<u>13.2 דרכים לפתרון</u>

אפשר למצות את תיאור המצב הקיים בשלושה משפטים:

- יותר ויותר צרכנים פונים אל יחידת ענ"א בבקשות לספק שירותים;
- יחידות ענ"א מתקשות להיענות לדרישות הגוברות בגלל העומס בתחזוקת התכניות הקיימות ובגלל עבודות פיתוח מתוכננות;
- קיים ציבור לא מבוטל בארגון, אשר מוכן <u>בעצמו</u> לטפל בבעיות, הדורשות פתרון מחשב;

מצב זה דורש פתרון אשר חייב לענות על הדרישות הבאות:

- הקלת העומס ביחידת ענ"א;

- מתן גישה ישירה, בלתי אמצעית אל המסוף של המחשב למשתמשי קצה רבים ככל האפשר;

- העמדת כלים מתאימים לרשות המשתמשים הנ"ל כדי להבטיח שירות טוב, מהיר ויעיל.

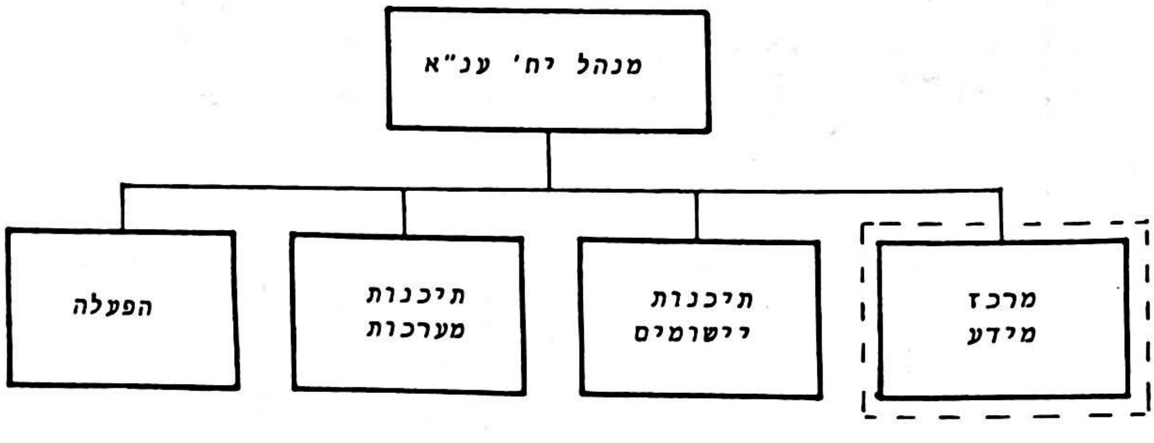
אחד מהפתרונות המוצעים הינו <u>מרכז המידע</u> (INFORMATION CENTER).

<u>13.3 מרכז מידע</u>

<u>מרכז המידע</u> (INFORMATION CENTER) הינו מערכת ארגונית, שמטרתה לאפשר תפוקת עבודה מקסימלית של <u>משתמשי קצה</u> (END-USERS), בתוך כדי מיזוג אופטימלי של חמרה, תכנה ואמצעי סיוע לתמיכה במטרה זו.

הבה נבחן את ההגדרה:

1. מרכז מידע <u>במערכת ארגונית</u> משובץ בדרך כלל בתוך יחידות ענ"א, כפי שזה מוצג בציור 3-13.



ציור 3-13 - מרכז מידע בתוך יחידת ענ"א

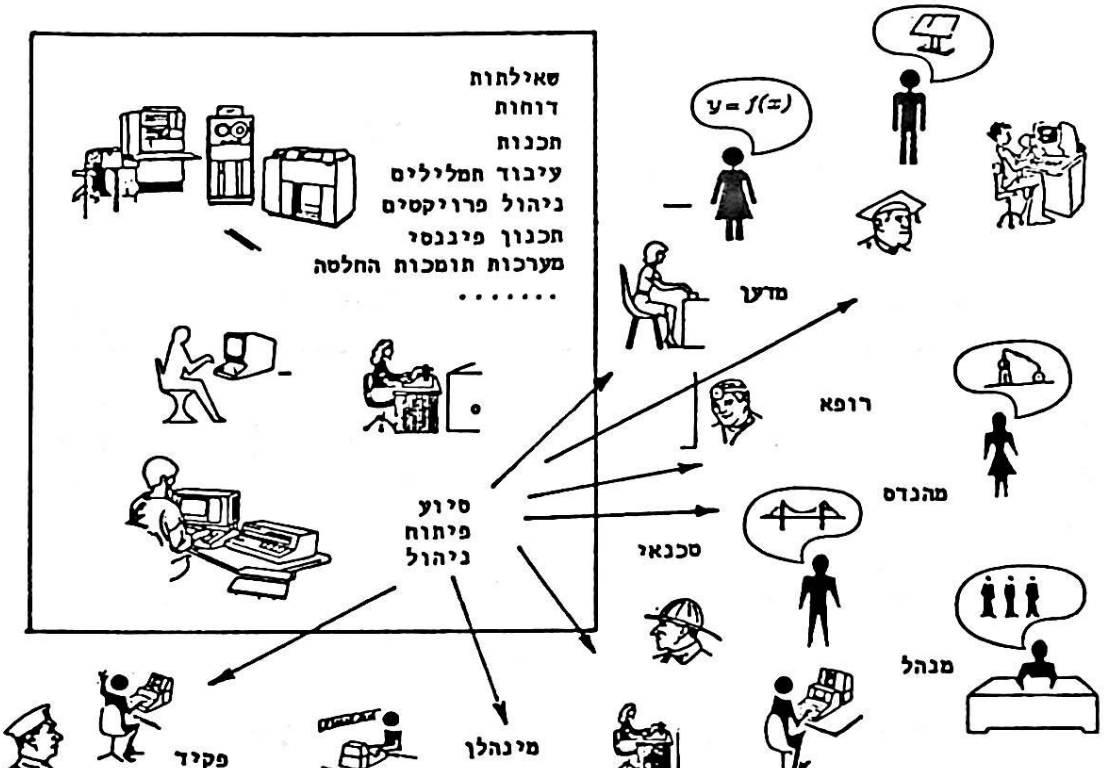
מרכז המידע כשלעצמו הינו מושג מופשט. במציאות הוא יכול ללבוש צורות רבות. <u>צורה אחת</u> יכולה להיות מבנה ובו מספר חדרים, כאשר ברוב החדרים נמצאים מסופים, המחוברים אל המחשב המרכזי; בחדר נוסף תימצא הספרייה, ובה ספרות מקצועית, ובחדר אחר יישב הצוות הקבוע של המרכז. צוות זה מורכב ממנהל המרכז ועוד שלושה עד ארבעה אנשים נוספים: אדם אחד יטפל באדמיניסטרצייה, אשר כוללת, בין היתר, גם מתן ססמאות למשתמשים, פרסום מידעון של המרכז בין עובדי

כל הארגון ובהקצאת משאבים. על אחד האנשים של יחידת ענ"א מוטל התפקיד של תפעול תקין ורצוף של המרכז, ואילו הנותרים באים להדריך את משתמשי הקצה, לסייע להם בעבודתם, ולפתח כלים נוספים בעבורם.

<u>צורה שניה</u> של מרכז המידע: המסופים מפוזרים על פני שטח המפעל כולו, אשר יכול להיות גדול מאד, כמו הקמפוס בטכניון. צוות המרכז, כפי שתואר קודם, נמצא בריחוק מן המשתמשים. הקשר אתם הוא בדרך כלל טלפוני, או כאשר המשתמשים באים אל המרכז, אם לא קיבלו פתרון לבעייה.

2. מי הם "משתמשי קצה", או בקצרה - משתמשים?

בשם זה אנו מתכוונים לכל אותם אנשים, אשר עובדים בארגון, ואשר מקצועם <u>איננו</u> קשור ישירות בעיבוד נתונים, אך לשם ביצוע עבודתם הם זקוקים לשירות של עיבוד נתונים. יתרה מזו - הם מוכנים עקרונית לעבוד על מסוף, אם זה לא יהיה מסובך מדי. אם הארגון בו מדובר הוא בית-חרושת, הרי הכוונה למהנדסים, מתכננים, פקידים, כלכלנים ועוד. אם הארגון הוא בית-חולים, אזי רופאים, אנשי המעבדה, עובדי מרפאות החוץ וכו' נכללים בקטגוריה זו. ציור 4-13 ממחיש באופן סכימטי, כיצד פועל מרכז מידע.



ציור 4-13 - מרכז מידע

הדמויות מחוץ למלבן מייצגות את משתמשי הקצה, ואלו שבפנים - את עובדי מרכז המידע, הממלאים את התפקידים, אשר תוארו בסעיף קודם.

3. **החמרה**, שמרכז המידע מעמיד לרשות המשתמשים היא למעשה רשת מסופים הכוללת מסכים, תווינים, מדפסות ואמצעי קלט ופלט אחרים, אשר יכולים לשמש את באי המרכז. מסופים אלה מחוברים אל המחשב המרכזי בין אם הם מפוזרים במפעל על פני שטח גדול, או בין אם הם מרוכזים במבנה אחד. בעזרת המסופים ניתן להתחבר אל הקבצים, או אל בסיסי הנתונים של הארגון, לשלוף ולעבד נתונים, כמובן בכפיפות לנוהלי עבודה של יחידת ענ"א.

4. לאחרונה פותחו **מוצרי תכנה** רבים, במיוחד בעבור משתמשי קצה. למוצרים אלו מספר תכונות משותפות:

- פשטות
- קלות הפעלה
- מסכים מנחים (תפריטים)
- הסברים קצרים
- פשטות באיתור ובתיקון טעויות

מוצרי תכנה אלו יכולים להתאים למגוון רחב מאד של המקצועות, שבהם דורשים משתמשי הקצה שירותים ממרכז המידע. חלק מהם פותחו ביחידות ענ"א, בעקבות הלחצים של המשתמשים. מנהלי מפעלים מצהירים היום בגלוי "קל יותר להכשיר מהנדס או כלכלן להיות מתכנת, מאשר להפוך תכניתן למהנדס או כלכלן...!"

נביא להלן שתי דוגמאות של מוצרי תכנה בעבור משתמשי קצה:

1. **מחולל דו"חות**

במקרים רבים זקוק משתמש לדו"ח קצר ופשוט. התכנה מאפשרת יצירת דו"חות שונים לאחר לימוד של 3-2 שעות בלבד. מחולל הדו"ח שולף את הנתונים מכל קובץ שהוא, שאליו קיבל המשתמש רשות גישה. שפת ההגדרות היא פשוטה ומובנת, וכוללת פונקציות מתמטיות שונות, המאפשרות לסכם שורות, עמודות, דפים שלימים, סיכומים חלקיים ועוד.

2. **תכנה לניתוחים סטטיסטיים**

כמעט בכל מפעל זקוקים היום למידע הקשור בניתוח סטטיסטי כלשהו. ניתוח מחקרי שווקים, הצגת מידע בצורה של עקומות או היסטוגרמות, חישובי ממוצעים מתקופות שונות והשוואתם - כל אלה מושגים בקלות רבה הודות לחבילות תכנה סטטיסטיות.

13.4 פתרון כוללני

גישה אחרת לפתרון הבעיות, אשר הוזכרו בראשית הפרק, אפשר למצוא במערכת ההידברותית של חברת יבמ: AS (APPLICATION SYSTEM).

נקודת המוצא של מפתחי מערכת זו היתה ההנחה, שבעלי מקצוע שונים נרתעים מלחפש פתרונות "ממוחשבים" לבעיותיהם בגלל "מחסום התכניתי". מפתחי המערכת טענו, שמנהל, איש עסקים או מדען זקוקים לעתים קרובות לפתרון מיידי של בעיות. פתרונות אלה ניתנים להשגה ע"י שימוש במחשב, אולם בעלי התפקידים הנ"ל נמנעים לבקש פתרון כזה בגלל החשש, שמא הצגת הבעיה לתכניתן, או הבנת פלט המחשב תתקלנה בקשיים מצדם, בגלל חוסר רקע מתאים במחשבים.

המערכת ההידברותית באה לגשר על הפער הזה ע"י מתן כלי פשוט, יעיל ומהיר למשתמש הלא-מקצועי בצורת מערכת שלימה של תכניות, אשר ניתנות להפעלה ע"י כל משתמש, בעברית או באנגלית. הסיסמא של מפתחי המערכת היתה "מחשב לכל עובד".

1. הפעלת המערכת

המערכת ניתנת להפעלה בכל מסוף, אשר מחובר למחשב שבו מותקנת תכנית AS. המחשב יכול להיות מותקן בארגון או במפעל, או לחילופין מחשב של חברת יבמ בלשכת השירות שלה, אליו ניתן להתחבר במסגרת שירות עיבוד מרחוק (REMOTE COMPUTING SERVICE). המשתמש פונה אל המערכת, לאחר שקיבל ססמה והרשאה, ונותן למערכת הוראות, מה ואיך לבצע. המערכת שואלת שאלות, ועל המשתמש לענות עליהן.

2. מבנה הנתונים

הנתונים מוצגים למשתמש בצורה פשוטה ביותר, ביצירת טבלאות דו-ממדיות, בדומה לדו"חות בעלי כותרות מתאימות, אשר מוכרים לכל עובד.

3. מגוון תכניות

המערכת מכילה ספרייה גדולה של תכניות מתחומים שונים, כגון סטטיסטיקה, תכנון פיננסי, ניתוח ומעקב אחרי פרויקטים ועוד. התכניות מוכנות להפעלה והמשתמש חייב לספק את המשתנים או הפרמטרים בלבד. הפעלת התכניות הללו ע"י המשתמש עצמו טומנת בחובה מספר יתרונות:

- המשתמש אינו חייב להכשיר את עצמו לעבודה בעיבוד נתונים, אינו חייב ללמוד תכנות, ואינו חייב להסביר לאדם אחר, למשל למנתח מערכות, מה הוא עושה במסגרת עבודתו, איך ולמה.

- המשתמש יכול לגשת אל הנתונים להם הוא זקוק, ישירות ובאופן מיידי. למיידיות נודעת השפעה עצומה, בעיקר אצל מקבלי ההחלטה בארגון, אשר חייבים לעמוד בתחרות מתמדת בעולם העיסקי הדינמי של ימינו.

- משתמשים מנוסים יכולים לבנות תכניות בעצמם, על-ידי שימוש במגוון רחב של פקודות, המיוחדות למערכת AS.

להלן תיאור קצר של התכניות (תת-מערכות) של AS.

1. תת-מערכת <u>MODEL</u> מיועדת לניתוח אסטרטגיות ולניתוחים פיננסיים. היא מאפשרת הגדרת יחסי גומלין בין מרכיבים שונים. ניתן לייחס תנאים שונים למודל, ולנתח אותם לאורך תקופות זמן.

2. <u>סטטיסטיקה וחיזוי</u> (STATISTICS/FORECASTING) - הינה תת-מערכת, הכוללת מבחר שיטות וכלים לניתוחים סטטיסטיים וטכניקות עיבוד וחיזוי כגון:

   - ממוצע, סטיית תקן, מתאם;
   - רגרסיות והתאמת עקומות;
   - סדרות עתיות;
   - מבחנים ואנליזות שונים.

3. <u>הצגה גרפית (GRAPH)</u> - מאפשרת הפקה של גרפים, היסטוגרמות, דיאגרמות פיזור וכו' בצורה קלה, באופן ישיר מבסיס הנתונים. במסוף צבעוני ניתן לשלוט עד שבעה צבעים, כולל כותרת, זיהוי צירים, קנה-מידה וכו'.

4. <u>ניהול ומעקב פרויקטים</u>

תכנית זו (NETWORK) משמשת לניתוח, פיקוח וניהול פרויקטים בשיטת <u>הנתיב הקריטי</u> (CRITICAL PATH). לאחר שהוגדרו יחסי הגומלין בין הפעילויות השונות, וציון משך הביצוע של כל פעילות, מצביעה התכנית על התופעות והתהליכים הבאים:

- המלצה למנהל הפרויקט, מה, מתי ואיך לבצע;
- ציון קשיים אפשריים;

* צבירת נתונים למעקב ולפיקוח.

בעזרת תכנה זו ניתן לטפל בהיבטים שונים של ניהול פרויקטים, כגון הקצאת משאבים, ניתוח סיכונים ושרטוט הרשת.

5. COMPOSE היא תכנית מיוחדת לעיבוד תמלילים. תכונותיה המגוונות כוללות עריכת שורות, עימוד, ספרור והחלפה של דפים, יצירה אוטומטית של תוכן עניינים ועוד.

6. תכניות עיבוד

קיימות מספר תת-מערכות המאפשרות עיבוד רגיל, יומיומי של הנתונים בעבור כל משתמש ומשתמש:

* UPDATE משמשת לעדכון קבצים.

* REPORT מאפשרת הפקת דו"חות בצורה מהירה ויעילה.

* DATA-ו EDIT הן שתי מערכות משולבות המאפשרות הזנת נתונים למערכת, בדיקת סדירותם, ביטולם ותיקונם באופן הדיבורותי.

הפעלת מערכת AS בארגון מביאה למעשה את עובדיה לשלב, שבו כל עובד יכול להשתמש בשירותי המחשב, ללא הכשרה וללא מתווך. כל אדם במפעל, אשר יודע להפעיל מסוף, יודע לעבד את הנתונים הדרושים לו, באופן מיידי ובשפתו הוא - בעברית. כך למעשה מגיעים אנו היום אל עידן חדש, בו הופך המחשב לכלי עבודה רגיל ומקובל, כמו הטלפון.

13.5 מלים לסיכום

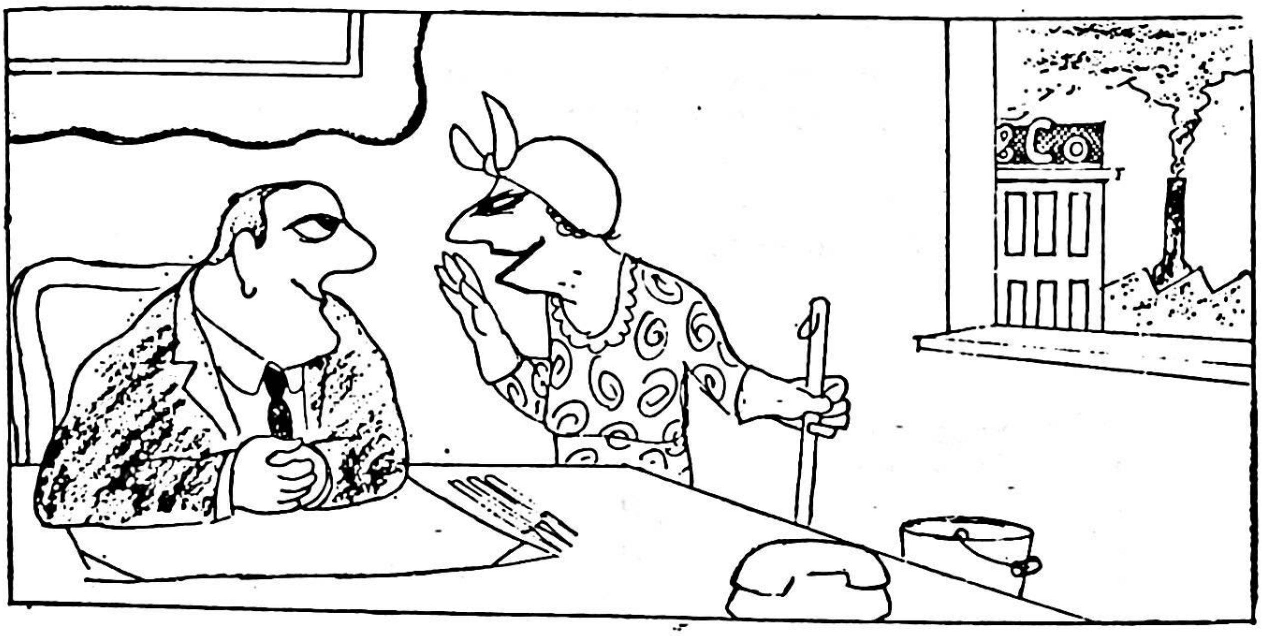
הסברנו את התפקיד והמרכיבים של מרכז המידע. אפשר עתה לנסות ולאפיין בצורה שונה את מרכז המידע: שיתוף קהילייית משתמשים במאגרי נתונים בסיוע כלי איחזור מידע. הנתונים הם מסוגים שונים וכן גם המשתמשים, אשר באים מכל מחלקות ותחומי הפעילות של הארגון.

ניתן לצפות כי מעורבות המשתמשים תרבה ותתרחב ומשתמש הקצה יהפוך, עם חלוף השנים, למיומן ולמתוחכם יותר בדרישותיו. המעורבות תהיה ביצירת הנתונים ועדכון שלהם ובדרישות איחזור המידע ועיבודו.

הדרישה האחרונה תכלול פיתוח כלי תוכנה שיאפשרו איחוד וצירוף נתונים מקבצים ובסיסי נתונים שונים ועיבוד שלהם גם בצורה גרפית, מחוללי יישומים המיועדים למשתמשי קצה, מעבדי תמלילים ושרותי דואר אלקטרוני. אוכלוסיות יעודיות יצפו לקבל שרותים מיוחדים כמו למשל מהנדסים, אנשי כספים, אנשי מינהלה וכדי.

חברה וארגון השואפים לגדול, לצמוח, להתרחב ולהתפתח, חייבים לדאוג לכך שמרכז המידע יוכר כיחידת שרות וסיוע. בצורה זו הוא יוכל לתרום בצורה המיטבית לכל צרכן או משתמש אשר מבקש לנצל את הנתונים ואפשרויות העיבוד שלהם לשם מילוי תפקידו בארגון.

פיתוח מרכז המידע והצלחת היישום שלו מותנים בתמיכה של ההנהלה, ובהכרה שזהו משאב חשוב ואמצעי שרות שיכול לתרום לצמיחת הארגון ולתמוך בהשגת מטרותיו.



מערכת מידע-מינהלי

פרק 13 - שאלות

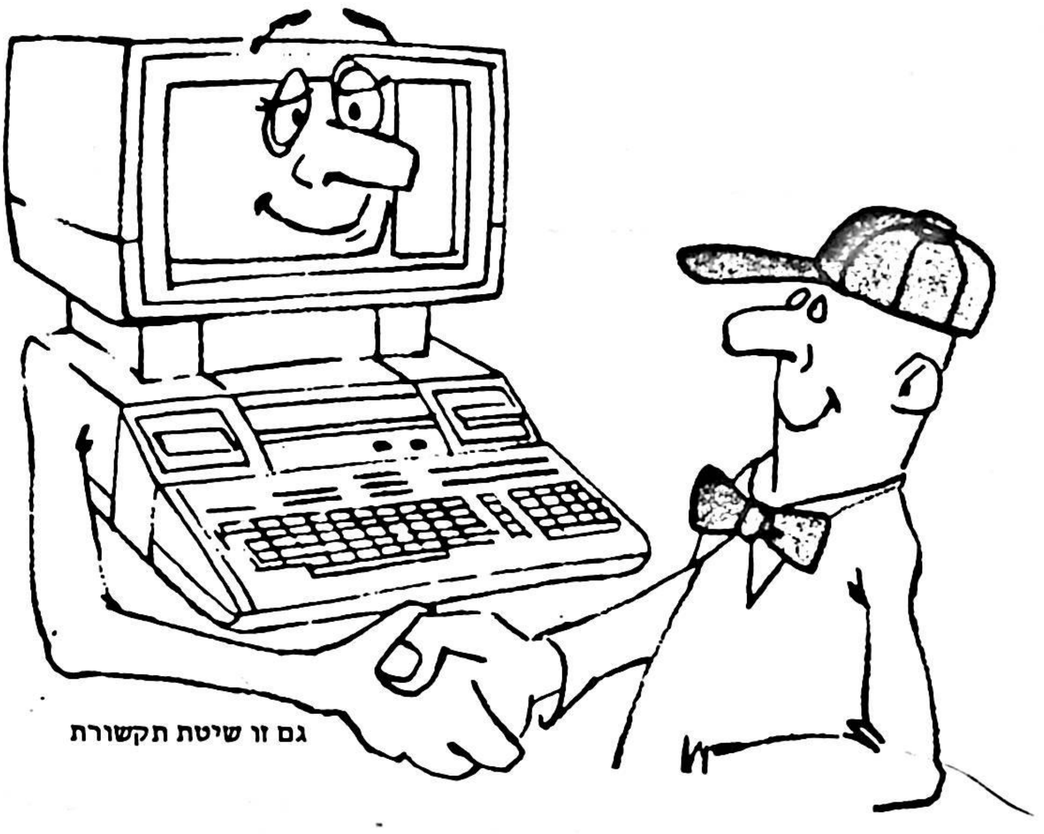
1. הסבר את הגורמים, אשר הביאו לפיתוח מרכזי מידע.

2. ציין סוגי תכנה, אשר לדעתך דרושים במרכזי מידע.

3. איזו שיטה של ניהול נתונים היית מציע למפעל, בו מקימים מרכז מידע. מדוע?

4. האם חשוב, שבכל מרכז מידע יהיה גם מחשב אישי, או ניתן להסתפק במסופים, המחוברים אל מחשב גדול? נמק תשובתך.

5. "מערכת AS של חברת יבמ מתאימה רק למוסד העוסק בסטטיסטיקה". הגב על ציטטה זו.



גם זו שיטת תקשורת

פרק 14 – תכנות בשפת בייסיק

אנו פוגשים במחשבים או בשלוחותיהם, המסופים, בכל אשר נפנה. נמצא אותם במשרד, במפעל, בבנק, בחנות הכל-בו, בבית הספר ואף בבית. המחשבים שונים זה מזה בגודלם ובאפשרויות העיבוד ואחסון הנתונים שלהם, אך בדבר אחד הם שוים - בחדירתם לתחומי פעילות חדשים, אשר לא ציפינו להם ובהיותם "בני לוויה" לנו.

הקשר שלנו עם המחשב נעשה עתה באמצעות מקלדת ומצג, אשר מקרין את הקלט והפלט של הנתונים. לפני שנים לא רבות היו הכרטיסים המגנטיים אמצעי הקשר היחיד של המשתמש במחשב או המתכנת עם המחשב, והפלט התקבל בהדפסה בלבד. ואילו כיום המצב שונה, שכן קיים קשר ישיר עם אמצעי העיבוד באמצעות המסוף או באמצעות המקלדת והמצג של המחשב (כשמדובר במיקרו-מחשבים). יש לכך יתרונות רבים ולא נעמוד עליהם כאן. החשוב שבהם הינו, כאמור לעיל, הקשר הישיר בין המשתמש לבין המחשב, אשר מאפשר תגובה מהירה על פעולות נדרשות, תיקון טעויות, אפשרות לבחור אלטרנטיבה לקבלת תשובה טובה יותר לבעיה, וזאת - במהירות ובדייקנות. שיטת העבודה במסופים העלתה את התפוקה של המשתמשים בהם ואת איכות העבודה שלהם.

את הנתונים המוזנים והמעובדים במערכת המחשב שומרים בסרט מגנטי, בקסטה, בתקליטון מגנטי או בדיסק, בהתאם לסוג המחשב, שפועלים בו, ובהתאם ליחידות ההיקפיות הקשורות אליו. לכל האמצעים האלו מטרה אחת בלבד: לקלוט את הנתונים למשמרת, כדי שאפשר יהיה להשתמש בהם פעמים נוספות, בלי שיהיה צורך לשוב ולהזין אותם. כדי שהנתונים יהיו בעלי משמעות ותקפים יש לעדכן אותם באמצעות קלט חיצוני או באמצעות תוצאות של פעולות חישוב שנעשו.

אמצעי הקלט של המחשב הוא המצג שמשמש להקרנת נתונים הנמצאים במאגרי הנתונים של המחשב ותוצאות חישוב או איחזור מידע. כדי לקבל דו"ח מודפס משתמשים במדפסת, אשר מפיקה תדפיסי דו"חות ותוצאות עיבוד שונות. מדפסות מיוחדות מאפשרות הדפסת גרפים או הדפסה בצבעים.

<u>תכנית מחשב</u>

אנו משתמשים במחשב כדי לקבל שירות, כדי לבצע במהירות משימה המוטלת עלינו: לפתור בעיה מתמטית, לקלוט נתונים ולעבד אותם להפקת דו"ח, לחשב משכורת או יתרות מלאי; לנתח אפשרויות שונות, כדי להחליט על דרך פעולה רצויה וגם... כדי להנות ולשחק במשחקי מחשב.

לא נוכל להציג כאן את כל האפשרויות הגלומות בהפעלת מחשבים, באלו דנו בפרקים אחרים. בפרק זה נבהיר כיצד אפשר להורות למחשב, מה לבצע, או כיצד לכתוב תכנית מחשב. התכנית הינה אוסף של הוראות, אשר ניתנות למחשב ע"י כותב התכנית, כדי שהמחשב יעשה בעבורו מה שהיה עליו לעשות, אילו עבד באמצעים ידניים בתקופה של טרום-מחשב.

מה אפשר לבצע באמצעות תכנית מחשב?

- להזין נתונים;
- לבצע חישובים שונים;
- להציג פלט במצג או במדפסת;
- לאחסן נתונים, לעדכן אותם ולהשתמש בהם בתהליכי עיבוד חוזרים.

את התכנית כותבים ב"שפה" המובנת למחשב, זוהי <u>שפת התכנות</u> (PROGRAMMING LANGUAGE). המחשב מכיר את המבנה המיוחד שלו או של קבוצת המחשבים שלה הוא שייך, ולכן התכנית, שהמחשב מבין, הינה <u>שפת מכונה</u> (MACHINE LANGUAGE). שפה זו קשה לשימוש בעבור מי שרוצה לתת הוראות למחשב, מכיון שהיא צמודה למבנה הפיסי שלו. לפיכך פותחו שפות אחרות, אשר נוחות יותר למשתמש ומאפשרות לו להציג את הבעיה, שהוא רוצה לפתור בשפה פשוטה יותר, אשר קרובה יותר לשפת אנוש מאשר לשפת המחשב. על פי רב אפשר למצוא שפות אחדות, אשר ניתן להפעילן במחשב נתון. יצרן המחשב מספק תכנית תרגום, אשר מעבירה את <u>תכנית המקור משפת התכנות אל שפת המחשב.</u>

<u>14.1 מהי שפת בייסיק ?</u>

בפרק זה נציג את התכנות בשפת <u>בייסיק.</u>

BASIC - BEGINNERS' ALL PURPOSE SYMBOLIC INSTRUCTION CODE

שפה זו פותחה בסוף שנות ה-60 כשפת לימוד הידברותית למתחילים ולתלמידים. בגלל פשטותה של השפה וקלות הלימוד מוצאים אותה היום בעולם המחשבים הקטנים, ובמיוחד בתחום ההוראה ובתחום המיקרו-מחשבים. היא מאפשרת לאנשים, שאינם תכניתנים, לתכנת בעצמם בעיות פשוטות ואף מורכבות בתחום עיבוד הנתונים המדעי והמסחרי, או להשתמש בה לתכנות משחקים ושעשועי מחשב.

שפת בייסיק בנויה לשימוש <u>הידברותי</u> (INTERACTIVE), כלומר היא מאפשרת למשתמש לכתוב הוראות מחשב, אשר הצירוף שלהן בסדר רצוי

יוצר את התכנית השלימה. בכך אין שפת הבייסיק שונה מכל שפה אחרת. השוני בינה לבין השפות האחרות הוא, בכך שהתכנית פועלת בשיטת הפענוח. בשיטה זו כל הוראה נבדקת מיד לאחר כתיבתה, והמחשב מבקש מהמתכנת לתקן שגיאות, אם ישנן כאלה, לפני שעוברים לכתיבת ההוראה הבאה. לאחר סיום כתיבת התכנית אפשר להפעילה מיד מן המסוף, להזין נתונים, לקבל תוצאות, או להפסיק את התכנית באמצע ע"י התערבות הנעשית מן המקלדת. את התכנית אפשר לאחסן לשימוש חוזר בספריית התוכניות, או בקסטה או בתקליטון הקשורים למיקרו-מחשב. תמיד נוכל לטעון את התכנית, שאוחסנה ולהשתמש בה פעם נוספת עם נתונים אחרים, שיוזנו לה. מן הראוי לציין, שיש שפות תכנות נוספות הפועלות בשיטה דומה.

פרק זה מכוון לערוך "פגישת היכרות" עם השפה ולהבהיר לך, הקורא, "מה" ו-"איך" אפשר לעשות בשפה זו. קורא, המעוניין להרחיב ולהעמיק את ידיעותיו בתכנות בייסיק, ימצא ספרות ענפה בנושא.

לפני שניגש ללימוד, הרי הערות אחדות.

1. שפת בייסיק הינה <u>שפת תכנות עילית</u> (HIGH LEVEL PROGRAMMING LANGUAGE), אשר לכאורה אינה תלויה במחשב מסוים. תכנית, אשר נכתבה בה, ניתנת להרצה בכל מחשב, שבו מופעלת השפה. הדבר נכון באופן חלקי. על השלד הבסיסי בנה כל יצרן תוספות והרחבות, ובכך יצר <u>ניב</u> חדש המתאים לשפה. הכינוי <u>בייסיק מורחב</u> (EXTENDED BASIC) מבהיר את העניין.

2. לפני שתשתמש בחומר המוגש כאן בדוק באיזו מידה הוא מתאים למחשב שבו תריץ את התכנית, ייתכן שיידרשו שינויים אחדים. במקומות שונים צוין, שישנם הבדלים, ויש לעיין בספרות של המחשב. במיוחד יש לשים לב לכללי ההפעלה של המסוף, לשימוש בלוח הקלידים ובשאר יחידות הקלט/פלט של המחשב.

3. הוראת הפלט העיקרית הינה PRINT, גם אם הפלט הוא תשובה במצג. כדי להשתמש במדפסת, יש לציין במפורש בהוראה, שאת הפלט יש להעביר למדפסת. בהמשך הפרק השתמשנו במלים "להציג" או "להדפיס", אך משמעותם זהה. פלט שמועבר למצג, יכול לעבור להדפסה באותו מבנה.

4. התכנות נעשה בשפה האנגלית ובאותיות גדולות בלבד.

5. אפשר לכתוב תכנית באנגלית, אשר תקרא נתונים, שמכילים מלל עברי. התכנית תוכל גם להפיק פלט עם מלל בעברית. נושא זה מורכב מעט ואף שונה ממחשב למחשב, ולכן לא עסקנו בו כאן. בדוגמאות השונות מוזן הקלט באנגלית והפלט גם הוא מוצג באנגלית בלבד.

לפני שנלמד את כללי התכנות, נעיין בתכנית קצרה, האם תוכל לפענח אותה?

```
10  REM THIS IS A SAMPLE PROGRAM
20  INPUT "TYPE TWO NUMBERS", A, B
30  C = A + B
40  IF C > 10 THEN GO TO 20
50  PRINT "C IS THE SUM OF A AND B", C
60  END
```

14.1.1 כללים בסיסיים

כמו לכל שפה, כך גם לשפת בייסיק יש "אל"ף-בי"ת" או מערכת תווים לשם בניית המלים, המספרים, ההוראות והביטויים השונים הנהוגים בשפה. האל"ף-בי"ת של בייסיק כולל את הסימנים הבאים:

אותיות : - אוסף בסיסי כולל את האותיות A עד Z.
- אוסף מורחב הכולל גם את שלושת התוים 0 , # , $.

ספרות : 0 , 1 עד 9.

סימני פעולות החשבון: + , - , * (כפל), / (חילוק),
↑ (העלאה בחזקה)

סימנים מיוחדים:

- נקודה תשמש לציון חלק עשרוני של מספר.
- פסיק ישמש להפרדה בין מספרים, מלים וכדי ואין לו משמעות של הפרדת חלקי מספר (כמו לציון אלפים למשל).
- סוגריים. יש רק סוג אחד של סוגריים, אשר משתמשים בהם לכל המטרות, שלא כמו באלגברה.
- סימן גרש (') וסימן גרשיים ("), ועוד.

לכמה מן הסימנים יש תפקידים מיוחדים, כפי שיובהר בהמשך.

14.1.2 הוראות בייסיק

1. תכנית בייסיק מורכבת מהוראות והגדרות שונות, אשר רשומות בשורות התכנית. כשם שיש לכל לשון כללי דקדוק וכתיבה משלה, כך גם כאן ישנם כללי כתיבה, שיש לקיימם, כדי להפיק תכנית נכונה, אשר תופעל במחשב. כאן המקום להזכיר, שהמחשב יבצע כל מה שהתכנית מבקשת גם אם אינה "הגיונית". היא תבצע, למשל, חיסור של שני מספרים במקום סיכום שלהם, אם כך נרשם.
   כל שורה של התכנית פותחת במספר סידורי, שאורכו עד 5

ספרות. סדר הביצוע של השורות יהיה לפי מספרן הסידורי, גם אם נכתבו שלא לפי הסדר הנכון. שורות אשר מכילות הגדרות לא תבוצענה.

מספרי השורות אינם חייבים להיות רצופים. אדרבא, רצוי לכתוב אותם בדילוגים של 10, למשל, כדי שאפשר יהיה להכניס שורות נוספות מאוחר יותר בלי לשנות את כל המספור. כאשר שורות חייבות להיות צמודות זו לזו, כמו במקרה של פקודות עוקבות, יש לתת להן מספרים סידוריים עוקבים.

2. כל הוראה נרשמת בשורה נפרדת והיא נכתבת משמאל לימין, כמקובל באנגלית. כתיבת ההוראות בשדה זה הינה חופשית למדי, ומאפשרת שימוש במרווחים בין מלים וסימנים. לשם רהיטות הקריאה אפשר, למשל, לכתוב A+B - 5 אך אי-אפשר לכתוב SU M (עם רווח בין האותיות), כי הן צריכות להיות צמודות כנדרש בשם של משתנה אלפביתי.

3. אורך השורה, כולל המספר, יכול להגיע ל-72 תווים בדרך כלל, אך יכולים להיות כללי כתיבה אחרים, אשר מגבילים את אורכה ל-40 תווים. כאשר פקודה גולשת משורה כלשהי, אפשר להמשיך אותה בשורה עוקבת, אשר מציינים בה סימן המשך. קיימות מערכות, בעיקר מערכות מיקרו-מחשבים, שבהן אין צורך בסימן המשך. ישנן מערכות, שבהן ניתן לכתוב מספר פקודות בשורה, ואז יש להפריד ביניהן באמצעות סימן מוסכם כלשהו, כמו : (נקודתיים) או \ (קו נטוי) וכדומה. לא מומלץ להשתמש בשיטה זו, אלא אם יש הכרח.

4. בסוף התכנית יש לרשום את הפקודה END, אשר מציינת, שזוהי <u>השורה האחרונה.</u> המספר הסידורי של שורה זו יהיה הגבוה ביותר בתכנית. הפקודה END יכולה למלא גם תפקיד של הוראת סיום של התכנית.

5. ניתן לסיים את התכנית ע"י הפקודה STOP, אשר תרשם בסיום המסלול הלוגי של ההוראות, גם אם אינה ההוראה האחרונה שתרשם בתכנית. משמעות ביצועה זהה להוראה END, כלומר, העברת הפיקוח למפעיל או למערכת ההפעלה.

<u>14.2 סוגי נתונים</u>

שפת בייסיק מכירה כמה סוגי נתונים: נתונים מספריים לפעולות חשבוניות ונתונים אלפנומריים, כלומר נתונים, שאין מבצעים בהם פעולות חשבוניות. הנתונים המספריים מתחלקים גם הם לקבוצות אחדות. נעיין עתה בצורות השונות להצגת נתונים בתכנית.

1. נתונים מספריים

מספרים שלימים (INTEGERS) הם מספרים בעלי ערכים שלימים בלבד, חיוביים או שליליים. אין צורך לציין סימן (+) למספר חיובי, אך יש לציין סימן (-) בעבור מספר שלילי. כאשר רושמים סדרת מספרים בשורה, מפרידים ביניהם בפסיק (,).

דוגמאות : 58
94, -389, 12835, 27

מספרים עשרוניים (FIXED-POINT) כוללים ערכים בעלי שבר עשרוני, והם יכולים להיות חיוביים או שליליים.

דוגמאות : .947
6.2, -37., 72.0

אפשר להרחיב את תחום הייצוג של המספרים ע"י שימוש במספרים בשיטת הנקודה הצפה (FLOATING POINT). שיטה זו נהוגה בעבודות מדעיות בלבד, ולא נדון בה בהמשך. צורת ההצגה הינה בעזרת חזקות חיוביות ושליליות של בסיס 10, למשל.

200000 ---> 2E5

2. נתונים אלפנומריים

סוג נתונים חשוב הינו הנתונים האלפנומריים, אשר משתמשים בהם כנתוני השוואה קבועים להדפסה ועוד. במחשב מיוצגים נתונים אלו כרצף של תווים: אותיות, ספרות, סימני פיסוק ועוד. רצף כזה של תווים יכול להיות מתו אחד ועד מקסימום של 255 תווים. שדה אלפנומרי תוחמים ע"י גרש או גרשיים, כפי שנקבע במחשב.

לדוגמה : 'THIS IS A PROGRAM'
"MODEIN STREET 100"

אורך השדה של הנתונים האלפנומריים הוא סכום כל התווים, הכלולים בין סימני התיחום (בין הגרשיים), כולל תווי רווח (SPACE).
לדוגמה :

קבוע אלפנומרי	אורך שדה
"XYZ"	3
"I AM A BOY"	10
"1539"	4

שים לב, לכך שהקבוע האלפנומרי האחרון מכיל ספרות, אך אי-אפשר להשתמש בו לפעולות חשבוניות אלא להשוואה ולהדפסה בלבד.

המתכנת יכול להשתמש בפונקצייה LEN, כדי למצוא את מספר התווים של שדה נתון, אם הדבר נחוץ.

אם סימן התיחום הוא גרשיים, ובתוך שדה התווים ישנו הסימן גרש, הוא ייחשב ככל תו אחר. אך אם ישנו בין התווים הסימן גרשיים, יש לחזור עליו פעמיים.

המשפט THE BOY'S HAT
ייכתב כך: "THE BOY'S HAT"
אורך השדה הוא 13 תווים.

המשפט THE "FUNNY" STORY
ייכתב כך: "THE ""FUNNY"" STORY"
אורך השדה הוא 17 תווים.

3. <u>משתנים</u>

תכנות עם מספרים קבועים אינו מאפשר גמישות ובמיוחד אינו מאפשר עיבוד עם נתונים משתנים. לפיכך משתמשים <u>במשתנים</u> (VARIABLES), אשר ערכם יכול, כשמם, להשתנות בעת הביצוע של התכנית. ההבחנה הזו זהה למקובל באלגברה.

המשתנים מייצגים, אם כן, ערכים משתנים של מספרים או תווים, המשמשים בתהליך העיבוד. למעשה הם מייצגים כתובות של תאי הזיכרון, שבהם נמצאים הנתונים או התוצאות של העיבוד. השימוש הסמלי בכתובת מאפשר להשתמש באותה תכנית עם סדרות שונות של נתונים.

שם משתנה יכול להיות מורכב מאותיות וספרות, אך הוא <u>חייב</u> להתחיל באות. בכל המערכות יש הגבלה של מספר התווים בשם המשתנה, מ-2 עד 7 תווים. כאשר אפשרי הדבר, רצוי להשתמש בשמות משתנים בעלי משמעות אשר יקלו על קריאת התכנית, כמו למשל SUM, NAME, QTY (עבור QUANTITY), LN (עבור LENGTH) ועוד.

כשם שישנן שתי קבוצות של קבועים, מספריים ואלפנומריים, כך יש שתי קבוצות של משתנים, בהתאמה.

<u>משתנה מספרי</u> כולל מספר בלבד בצורה אשר מאפשרת ביצוע חישובים. כותבים אותו כך: A, SUM, MONE, K3B. ישנן מערכות המאפשרות להגדיר משתנה מספרי, אשר מכיל ערך שלם בלבד, ללא הנקודה העשרונית. כותבים את השם בצירוף הסימן % (אחוזים), למשל: A%, SUM%. כאשר נעביר למשתנה כזה את הערך 7.6, למשל, נמצא בו 7 בלבד והחלק העשרוני יישמט.

בתחילת הביצוע של התכנית מתאפסים כל המשתנים המספריים

באופן אוטומטי. ערכם יישאר אפס עד אשר יוכנס אליהם ערך כלשהו. במהלך התכנית אפשר לאפס כל משתנה בפקודה, למשל A=0.

<u>משתנה אלפנומרי</u> מכיל תווים שונים: אותיות, ספרות וסימנים מיוחדים, אשר מיועדים להשוואה או להדפסה אך לא לחישוב. רושמים את שם המשתנה בצירוף הסימן $ (דולר), לדוגמה, D$, NAME$.
על פי רב מוקצב לכל משתנה אלפנומרי שדה באורך קבוע. בכמה מערכות ארכו 14 תווים. אם נכתוב, למשל, "I LOVE BASIC" = LET A$, אורך המחרוזת A$ הוא 12 תווים, ולכן יהיו בשדה מימין 2 תווי רווח, כדי להשלים ל-14. אם המחרוזת ארוכה יותר, יושמטו התווים מימין.

אפשר להגדיר משתנה בהוראה DIM. למשל, כדי להגדיר משתנה K$ באורך 18 תווים נכתוב: 500 DIM K$18.

<u>14.3 פעולות חשבון</u>

פעולות החשבון המוכרות לבייסיק הן הפעולות האלגבריות הרגילות: חיבור, חיסור, כפל, חילוק והעלאה בחזקה. סימני הפעולה שונים מהרישום המקובל באלגברה:

סימול בייסיק	משמעות	דוגמאות	
+	חיבור	6+R	A+B
-	חיסור	7-4	K-3
*	כפל	9*5	R*8
/	חילוק	7/8	Z/9
↑ או **	העלאה בחזקה	2**3	I**7
		(2^3, I^7)	

לא קיימים סימנים מיוחדים להוצאת שורש, למשל. פעולה זו תוצג על ידי פעולות העלאה בחזקה כמו B**0.5, וזאת כדי לבטא שורש ריבועי של B. פעולה זו אפשר לבטא גם על ידי הפונקציה SQR(B). פעולה כמו אינטגרל, סינוס או סכום סדרת מספרים יוצגו על ידי פונקצייה מתאימה, כפי שיוסבר בהמשך.

<u>תרגיל</u>

עדיין אין בידנו כל הכלים לכתוב תכנית שלימה, אך עד עתה למדנו די, כדי לנסות לכתוב תכנית פשוטה. מטרתה להראות, שאפשר לומר

למחשב באמצעות התכנית מה לעשות, אך עלינו לדעת תחילה מה רוצים לבצע: כיצד להזין את הקלט, וכיצד לבצע את החישובים ואיך לקבל את הפלט. כמה מההוראות שיינתנו בתרגיל יוסברו בפירוט רב יותר בהמשך. אך הבה נתחיל.

אורי עזר לאמו והלך למכולת לקנות כמה מוצרים. בשובו ערך את חישוב הערך של הקנייה, ובדק, אם קיבל את העודף הדרוש. הקנייה כללה 2 מוצרים, מוצר א׳ במחיר 52 שקל ומוצר ב׳ במחיר 38 שקל. אורי זכר, שמסר לחנווני שטר של 100 שקל. כמה עודף קיבל?

אורי החליט ללמוד בייסיק ולכתוב תכנית, למרות שהבעייה נראתה לו פשוטה מאד:

`10 LET A = 52 + 38`	סכום הקנייה
`20 LET B = 100 - A`	העודף
`30 PRINT A,B`	הדפסת התוצאה
`40 END`	סוף התכנית

עתה הבין, שכאשר ירצה לחשב סכום קנייה ועודף במקרה אחר, יהיה עליו לכתוב את התכנית מחדש. ומה יקרה, אם יהיו פריטים רבים יותר, וסכום הכסף שיקבל מאמו, יהיה שונה?

הוא החליט לכתוב תכנית כללית ל-5 פריטים. כאשר יקנה פחות מ-5, ירשום 0 במקום ערך הקנייה של הפריטים, שלא קנה. את סכום הכסף, שקיבל לקנייה הוא יזין בהתחלה, כנתון ראשון. התכנית שלו היתה שונה עתה. הוא גם למד, שאפשר לוותר על כתיבת המלה LET.

```
10 INPUT MONEY,A,B,C,D,E
20 SUM = A+B+C+D+E
30 CHNG = MONEY - SUM
40 PRINT "PURCHASE AMOUNT=",SUM
50 PRINT "CHANGE=",CHNG
60 END
```

נסקור עתה את הוראות התכנית ונבין את משמעותן:

10 הוראה לקלט של 6 מספרים. המספר הראשון יהיה בעבור סכום הכסף, שאורי יקבל לקנייה, והמספרים האחרים יהיו ערכי הקנייה של הפריטים השונים. אם יקנה 4 פריטים למשל, הוא ירשום אפס במקום המחיר של הפריט האחרון. אורי ידע, שעליו להזין נתונים, כאשר התכנית תדפיס סימן שאלה (?) ותחכה.
שורת הקלט שלו תהיה: 100,20,15,40,0,0 (משמאל לימין כמובן).

20 הוראה זו מורה למחשב לסכם 5 מספרים ולשמור את התוצאה במשתנה SUM.

30 הוראה לחישוב הפרש של שני מספרים ואת התוצאה לשמור במשתנה CHNG.

50,40 אלו הן שתי הוראות זהות כמעט, להדפסה של סכום הקנייה (הוראה 40) וסכום העודף (הוראה 50). כדי שאורי ידע מה הודפס (או הוצג במסך), הוא החליט להדפיס קבוע אלפנומרי להסבר. את הקבוע שיודפס הוא הקיף בגרשיים, ולאחריו כתב את שם המשתנה, אשר ערכו יודפס.
בהנחה, שקיבל לקנייה סכום של 1000 שקל, והקנייה היתה בסכום כולל של 750 שקל,
תהיה ההדפסה :

```
PURCHASE AMOUNT = 750
CHANGE = 250
```

60 כאן הסתיימה התכנית, ואורי החליט להרחיב את ידיעותיו בשפת בייסיק.

14.3.1 ביטויים אריתמטיים

כללי הבנייה של ביטויים אריתמטיים דומים מאד לכללי הבנייה של ביטויים אלו באלגברה. נתבונן במספר ביטויים, שנכתבים בכתיב המתמטי המקובל ובבייסיק. בדוגמאות אלו המשתנים מייצגים נתונים מספריים בלבד. בכל מקום שאפשר לרשום משתנה אפשר לרשום פונקצייה או ביטוי שלם אחר.

בייסיק	כתיב מתמטי מקובל
A+B-3*C	$A+B-3C$
A**2+B**2	A^2+B^2
(A+B)**0.5	$\sqrt{A+B}$
(A+2)/(B*C)	$\frac{A+2}{B.C}$

אפשר להשאיר <u>תווי מרווח</u> (BLANK) בין משתנה לבין סימן הפעולה או לבין הסוגריים, כדי להקל על קריאת הביטוי. על משמעות כתיבת הסוגריים נעמוד בהמשך.

14.3.2 הערכת ביטויים

ערכם של ביטויים אריתמטיים נקבע לפי כללי האלגברה, ולפי הסדר של רישום הפעולות השונות, משמאל לימין: סוגריים, העלאה בחזקה, כפל וחילוק, חיבור וחיסור.

נתון :
דרוש : A=5, B=2, C=7
חישוב : B-3*C

שלב 1 3*C = 3*7 = 21
שלב 2 B-21 = 2-21 = -19

התוצאה: -19
דרוש :
חישוב : A/B*C

שלב 1 A/B = 5/2 = 2.5
שלב 2 2.5*C = 2.5*7 = 17.5

התוצאה: 17.5

הפעולות בעלות אותה עדיפות תבוצענה משמאל לימין, ולפיכך הביטוי A/B*C, למשל,
מתאים לביטוי $\frac{A\cdot C}{B}$ בכתיב המתמטי המקובל.

הביטוי A-B/C שווה בערכו לביטוי A-(B/C).
אבל כדי לחשב את $\frac{A}{B\cdot C}$ יש צורך להשתמש
בסוגריים, כדי לשנות את סדר העדיפות של הפעולות בביטוי ולבצע קודם מכפלה, B.C ואח"כ חילוק של A במכפלה.
כלומר, $A/(C*B) = \frac{A}{B\cdot C}$

אם יש ביטוי הכולל "סוגריים בתוך סוגריים", יש לחשב תחילה את הביטוי שבסוגריים הפנימיים ואח"כ את זה שבסוגריים החיצוניים יותר וכן הלאה, כמקובל באלגברה. שים לב, שיש רק סוג אחד של סוגריים, ולא כמקובל באלגברה.

כאשר רוצים לרשום מספר שלילי, יש לרשום "מינוס" משמאל ולהקיפו בסוגריים, לדוגמה :
6*(-B) או 5+(-3)

ביטויים אלפנומריים כוללים צירוף או השוואה, אך אי אפשר לבצע בהם פעולות חשבוניות. אפשר, למשל, "לחבר" שני משתנים אלפנומריים, כלומר לצרף את התווים של המשתנה הראשון אל המשתנים של המשתנה השני. לדוגמה: אם משתנה X$ מכיל את המלה TEL- והמשתנה Y$ מכיל את המלה AVIV, הרי החיבור X$+Y$ ייתן לנו את TEL-AVIV. שים לב, לכך שמחרוזות התווים תהיינה צמודות. אם דרוש רווח ביניהן, יש לדאוג לכך.
בהמשך נציג פעולות שונות בביטויים אלפנומריים.

14.4 מושג השוויון והוראת ההצבה - LET

בעבודה במחשב יש לסימן השוויון שני תפקידים: התפקיד הראשון **בהוראת הצבה** (ASSIGNMENT STATEMENT) והשני - **בהוראת בקרה**

(CONTROL STATEMENT). נלמד תחילה את הוראת ההצבה.

<u>הוראת הצבה</u> "מעבירה" ערך מספרי למשתנה כלשהו. מבנה ההוראה דומה למשפט השוויון האלגבראי:

לדוגמה: ביטוי אריתמטי = משתנה

```
M = 5
A = B+C*3
```

בצד שמאל של סימן השוויון ירשם <u>משתנה אחד בלבד</u>, ובצד ימין יכול להיות ביטוי אריתמטי, קבוע או משתנה. כאשר מימין לסימן השוויון יש קבוע או משתנה, יועבר הערך שלהם אל המשתנה שמשמאל לסימן השוויון, <u>במקום</u> הערך שהיה בו קודם לביצוע הפעולה. כאשר יש ביטוי, ערכו יחושב והתוצאה תועבר למשתנה.

הפירוש של מושג השוויון (=) הינו: חשב את ערך הביטוי שבצד ימין לסימן השוויון והצב ערך זה למשתנה המופיע בצידו השמאלי של סימן השוויון. אם נתון, למשל, משתנה I שערכו 7 ונתונה ההוראה K=I+3, תתבצע הפעולה הבאה: חישוב הביטוי I+3 והתוצאה 10 תועבר למשתנה K <u>במקום</u> הערך הקודם שהיה בו.

כאשר נתון לדוגמה: SUM=5

ונחשב SUM=SUM+2

הפעולה הראשונה תהיה חישוב SUM+2 והתוצאה היא 7. הערך 7 יועבר למשתנה SUM ויחליף את הערך הקודם, שהיה בו. שים לב, שהחישוב נעשה באוגרים, ורק התוצאה מועברת לזיכרון. הפעולה, שהודגמה זה עתה, הינה "פעולת סיכום למונה".

באמצעות משפט הצבה, שמשתנה זהה מופיע בשני צדיו, אפשר לייצג בתכנית שימוש במונה. במושג זה מתכוונים למשתנה שערכו משתנה לאחר כל פעולה. המונה יכול להתקדם ביחידה אחת או יותר, בהתאם לעניין. למשל, כדי "לספור" את מספר הפעמים, שחזרנו על פעולה מסויימת נקדם את המונה ביחידה אחת לאחר כל פעולה. נוכל לרשום זאת כך: 10 K = K+1.

בכל פעם שתתבצע שורה 10, יעלה ערכו של K ב-1. אם ערכו היה 8 <u>לפני</u> הביצוע, יהיה ערכו 9 (או 8+1) <u>לאחר</u> הביצוע. מונה זוגי יירשם כך:

```
30 M = M+2
```

<u>14.4.1 איך מאפסים משתנה?</u>

כאשר רוצים לאפס את מונה B, עושים זאת ע"י הצבת הערך אפס, וכותבים זאת כך:

```
15  B=0
```

לאחר הפעולה יהיה ערכו של המשתנה B אפס בלי להתחשב במה שהיה בו קודם לביצוע ההוראה.

בכמה מהניבים של בייסיק חובה להקדים את הוראת ההצבה במלה LET. אך בדרך כלל אין צורך בכך. לדוגמה: 12 LET C = 5. המלה LET נותנת, כמובן, נופך "אישי" וסגנון של שפת בני-אדם, אך ברב המערכות אין צורך להשתמש בה.

דוגמאות להוראות הצבה בתכנית:

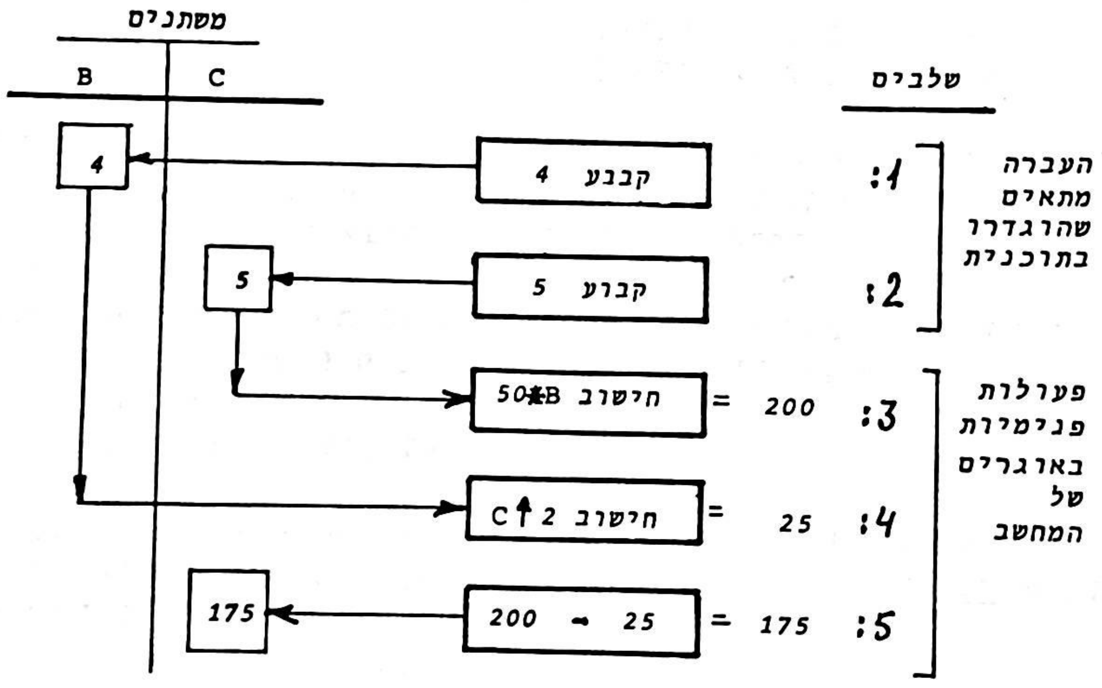
```
11 B = 4
12 C = 5
13 C = 50 * B - C**2
21 P$ = "THIS IS A BOY"
22 W$ = P$
```

מה ערכם של C ושל W$ בסיום הפעולה?

- ערכו של C יהיה 175, וערכו של W$ יהיה שווה ל-P$.

ומה קרה ל-B ול-P$? - הם <u>נשארו ללא שינוי</u>, כי פעולת ההצבה מהם אל משתנה אחר דומה לפעולת "העתקה" ולא לפעולת "העברה" פיסית של עצם ממשי ממקום למקום.

נתאר בתרשים את סדרת ההוראות 11, 12, 13.



<u>תרגיל</u>

נוכל עתה לפתור משוואה ריבועית, משימה שאינה אהובה ביותר על תלמידים רבים:

המשוואה : $AX^2 + BX + C = 0$

נתון : C=-1 ,B=3 ,A=2

הנוסחה לפתרון המשוואה :

$$X_{1,2} = \frac{-B \pm \sqrt{B^2-4AC}}{2A}$$

את הביטוי מתחת לסימן השורש מקובל לכנות "דלתה". נפשט מעט את התרגיל ולא נתייחס למצבים, שבהם דלתה שווה לאפס או לערך שלילי. הסימנים X1, X2, המשמשים אותנו באלגברה לסימון שורשי המשוואה, אינם מקובלים בבייסיק, ועל כן אפשר להשתמש בסימון X1, X2, או בכל סימון אחר, למשל R, S1, S2 ועוד.

את הנוסחה נפתור בחלקים, כמקובל באלגברה: נציג את ערכי המשתנים, נחשב את הערך של "דלתה" ואח"כ נחשב את הערכים של המשתנים.

קביעת ערך למשתנה A	`10 LET A=2`
קביעת ערך למשתנה B	`20 B=3`
קביעת ערך למשתנה C	`30 C=-1`
חישוב דלתה	`40 D=(B**2-4*A*C)**0.5`
חישוב X1	`50 X1=(-B+D)/(2*A)`
חישוב X2	`60 X2=(-B-D)/(2*A)`

הוראות התכנית מציגות בשלבים את דרך הפתרון, והן מבוצעות בזו אחר זו לפי סדר כתיבתן. ההוראות המוצגות כאן מכוונות לפתרון המשוואה, ואינן מייצגות עדיין תכנית שלמה. בשלוש השורות הראשונות קובעים את הערכים של מקדמי המשוואה הריבועית למשתנים C,B,A. בשורה 40 מחשבים את הביטוי "דלתה" וקובעים את ערכו למשתנה D. ולבסוף בשתי השורות האחרונות, מחשבים את שני השורשים של המשוואה ומאחסנים את התשובות במשתנים X1, X2 בהתאמה. וזהו סוף פשוט, לא כן?

שים לב, שסוגריים חובקים את הביטויים שלפני סימן החילוק ושאחריו (משפטים 50, 60), כדי שהביטויים יחושבו, <u>לפני</u> שתבוצע פעולת החילוק, כפי שנדרש בנוסחה.

<u>14.5 משפט הגדרת נתונים - DATA ופקודת הקריאה - READ</u>

כדי לפשט את הצבת הנתונים למשתנים, אפשר להשתמש בהוראת DATA

להגדרת הנתונים. אפשר לכתוב, למשל:

```
100 DATA 15 ,7 ,25 ,19 ,12
```

או גם כך, בשורות רצופות:

```
101 DATA 15 ,7 ,25
102 DATA 19 ,12
```

כדי שהתכנית תשתמש במספרים אלו, יש להשתמש בפקודת קריאה READ:

```
300 READ A ,K ,R ,M ,E
```

READ תגרום לכך שהמשתנים E,M,R,K,A יקבלו את הערכים 15, 7, 25... 12, בהתאמה. כלומר תתבצע פעולת הצבה במשתנים אלו כתוצאה של פעולת "קריאה". המשתנה A יקבל את הערך 15, K יקבל את הערך 7 וכן הלאה, ו-E יקבל את הערך 12.

פקודות הקריאה יכולות גם להיות נפרדות. בכל פעם תקרא התכנית את המספר התורן (שנמצא בתור) מתוך הוראת DATA, והיא תגלוש משורה לשורה, אם דרוש.

```
420 READ A ,K
510 READ R
560 READ M ,E
```

גם במקרה זה יקבלו המשתנים את הערכים בהתאמה, לפי סדר כתיבתם בהוראת DATA. אם רוצים להשתמש בערכי הוראת DATA פעם נוספת מתחילתם, גם אם נקראו כולם, יש לכתוב את ההוראה RESTORE.

לדוגמה:

```
100 DATA 8, 4, 7
110 READ K, B, S
120 RESTORE
130 READ X, Y, Z
```

המשתנים S,B,K יקבלו את הערכים 7,4,8 בהתאמה וכן גם המשתנים X, Y, Z.

נתונים אלפנומריים יירשמו בהוראת DATA, כאשר הם "חבוקים" בגרשיים (או גרש).

```
15 DATA 1948
16 DATA "ISRAEL" ,1983
30 READ A, R$
```

ההוראה READ בשורה 30 תגרום לקריאת הקבוע 1948 אל המשתנה A ולקריאת התמליל "ISRAEL" אל המשתנה R$. הקבוע 1983 ייקרא בהוראת READ הבאה. אם נוסיף, למשל, את השורה 50 READ A לתכנית, יכיל המשתנה A את הערך 1983 במקום הערך 1948, אשר היה בו קודם, לאחר ביצוע שורה 30.

הוראות DATA ניתן לכתוב בכל מקום בתכנית, גם לאחר הוראות

READ. המחשב מתייחס לכל ההוראות כאל רצף נתונים אחד לפי סדר הופעתן בתכנית, כלומר, לפי מספרן הסידורי.

14.6 קלט ופלט של נתונים

בהוראת הצבה משתמשים, כדי לקבוע ערכים למשתנים, שנדרשים בהמשך התכנית. כאשר ישנה סדרת נתונים ארוכה, אין השיטה נוחה ומשתמשים במשפט DATA. הנתונים בהוראת ההצבה ובהוראת DATA נכתבים בעת כתיבת התכנית, ואינם ניתנים לשינוי בעת הביצוע. אפשר לתקן את התכנית ולהריץ אותה שנית, אך מרגע שהתחילה לפעול, אין אפשרות להתערב ולשנות נתונים. ההוראות תתבצענה בתוך כדי שימוש בנתונים, שנרשמו לפני שהתכנית החלה לפעול, כלומר <u>בנתונים קבועים</u>.

קביעה מראש של נתונים טובה לעניינים מסויימים, כמו גורמי השוואה, אך אינה מאפשרת עבודה בדו-שיח עם המחשב. המשתמש רוצה להזין נתונים לתכנית, להגיב על תשובות ותוצאות שקיבל ולהזין נתונים אחרים וכן הלאה, עד לקבלת התוצאה הרצויה, או עד שהוא מחליט להפסיק את העיבוד.

חיסרון נוסף הוא, שקיבולת הזיכרון אינה מאפשרת לרשום נתונים רבים בתוך התכנית. לפיכך יש להזינם ע"י המפעיל, אם הם נתונים חדשים, או לשמור אותם בקבצים מגנטיים ולקרוא אותם לזיכרון לפי הצורך, במנות קטנות. בנושא זה נדון בהמשך.
הזנת הנתונים על ידי המפעיל נעשית באמצעות הוראת הקלט INPUT.

14.6.1 הוראת קלט - INPUT

מבין הדרכים האפשריות למסירת נתונים לתכנית, תמשוך הוראת INPUT את תשומת לבך. היא תאפשר לך להזין נתונים לתכנית בעת הביצוע, על פי דרישת התכנית. כיצד תדע, שהתכנית מבקשת ממך להזין נתונים? התכנית תדפיס סימן שאלה (?), כאשר היא מבצעת את הוראת INPUT, ולאחר מכן תהיה מוכנה לקלוט את הנתונים לפי הצורך.

ההוראה

```
20 INPUT A, B, X$
```

תגרום להדפסת סימן שאלה (?), והתכנית תחכה להזנת 3 שדות נתונים, שיופרדו בפסיק. שני השדות הראשונים, שיוזנו למחשב, צריכים להיות מספריים (שישמשו לחישוב) והשדה השלישי הוא אלפנומרי, אשר יכול להיות מספר גם הוא (למשל, מספר הבית ברחוב, שבו אתה גר), אך הוא לא ישמש לחישובים. לאחר גמר הפעולה תיראה במסך השורה הבאה:

```
? 583, 26, "DANNY"
```

השדה הראשון שלאחר סימן השאלה הוא מספר זיהוי כלשהו, השני – גיל, והשלישי – שם. בין השדות מפרידים בפסיק, ואפשר להוסיף רווחים, כדי שלא לכתוב בצפיפות. ישנם מחשבים אשר בהם אין צורך לרשום גרשיים לנתון כאשר קוראים אותו למשתנה אלפנומרי. המערכת תזהה עובדה זו ותבצע את הפעולה הנכונה.

יש להזין בדיוק את מספר השדות הנדרש על ידי התכנית. אם רושמים פחות, התכנית "תחכה" להזנת כל השדות, פרט לניבים של השפה, שבהם אפשר להזין קו נטוי (/) כדי להפסיק את ההזנה. ואם מזינים יותר, על פי רוב תדרוש התכנית להזין את הכל מחדש, אך במקרים מסויימים היא יכולה פשוט להתעלם מהשדות המיותרים. אם רוצים לדלג על הזנת ערך כלשהו, אפשר לכתוב שני פסיקים רצופים, למשל, כדי לדלג על הזנת "גיל" כותבים "DANNY",,582. כתוצאה מכך הערך של משתנה B בתכנית לא ישתנה.

הזכרנו קודם, שישנה בעיה מסוימת ב"פענוח" סימן השאלה המודפס על ידי התכנית: מהם הנתונים, שהתכנית מבקשת כעת (כאשר היא עשויה לבקש נתונים שונים) וכמה ערכים יש להזין. כדי להקל על המשתמש, אפשר לתת לו הנחיות לפני כל הוראת INPUT, כמו למשל:

PLS TYPE YOUR ID, AGE, NAME

אפשר להציג הנחייה זו באמצעות הוראת הפלט PRINT, או שאפשר לשלב את הצגת ההנחייה בהוראת INPUT. בהוראת INPUT יש לרשום נקודה-פסיק(;) לאחר ההודעה, ופסיקים בין המשתנים שברשימת השדות לקליטה.

לדוגמה: `20 INPUT "TYPE YOUR ID,AGE,NAME";A, B, X$`

התכנית תדפיס `TYPE YOUR ID,AGE,NAME?`

סימן השאלה יופיע לאחר התו האחרון, ומיד לאחריו יוזנו שדות הנתונים שהתכנית מבקשת.

כאשר מציגים את ההודעה באמצעות הוראת PRINT, יודפס סימן השאלה בשורה נפרדת, כיון שהוא לא שייך להוראה שנתנה את ההודעה. נראה זאת בהמשך.

14.6.2 הוראת פלט - PRINT

הוראת הפלט PRINT מקבילה להוראת הקלט INPUT. האחת קולטת נתונים אל המחשב, והאחרת מאפשרת העברתם מזיכרון המחשב החוצה, לשימוש בני אדם או לשם רישום ביחידת אחסון. לדוגמה:

ההוראה `20 PRINT "SHALOM"`

תגרום להדפסה `SHALOM`

ההוראה `60 PRINT Y$,R,4`

תגרום להדפסת הערכים של המשתנים R, Y$ ולהדפסת המספר 4. ההדפסה תבוצע בשורה אחת ברווחים סטנדרטיים של 14 טורים לכל שדה (או מספר אחר - בניבים שונים של בייסיק). אורך השורה המקסימלי - 72 תווים.

לדוגמה, אם

```
R = 87
Y$ = "ISRAEL"
```

נקבל לאחר ביצוע שורה 60 את ההדפסה הבאה:

מספר הטור	42	28,29	14,15	1
הערך שיודפס	4	87		ISRAEL

אם אורך שורה אחד עובר על 14 טורים, יודפס השדה הבא בטור הבא הפנוי. משתנה אלפביתי יוצב בצד שמאל של השדה (קבוצת 14 הטורים), ומשתנה נומרי יוצב בצד ימין של השדה.

כדי להדפיס את הנתונים צמודים זה לזה ולא בטורים קבועים, יש לרשום בין המשתנים או הקבועים בהוראת ההדפסה נקודה-פסיק(;), למשל:

```
75 PRINT "YOUR CHANGE IS:" ;A
```

ובהנחה שהערך של A הוא 25 נקבל:

```
YOUR CHANGE IS:25
```

כל הוראת PRINT תגרום להדפסה בשורה חדשה. אם רוצים להדפיס ברצף, יש להוסיף סימן פסיק(,) או נקודה-פסיק(;) בסוף רשימת הערכים בכל הוראה פרט להוראה האחרונה. למשל, כדי להדפיס "TYPE YOUR NAME" נכתוב:

```
15 PRINT "TYPE YOUR";
16 PRINT "NAME"
```

והתוצאה

```
TYPE YOURNAME
```

כמובן, ללא רווח בין שתי המלים, מכיון שלא השארנו תו רווח. את השורה 16, למשל, צריך לכתוב כך:

```
16 PRINT " NAME"
```

שים לב לכך שלשתי השורות מספרי שורה עוקבים.

ראינו שבהוראת הדפסה ישנם שמות המשתנים להדפסה, ואם דרוש - גם תמליל קבוע. אפשר לכלול בה גם ביטוי, אשר יחושב ויודפס מיד, לדוגמה:

```
120 PRINT M,K*3
```

תרגיל

עתה נכתוב תכנית, אשר תקלוט שלושה מספרים כלשהם ותחשב:

- ממוצע של שלושת המספרים.
- מנת החילוק של המספר הראשון במכפלת המספר השני בשלישי.

התכנית תבקש מן המפעיל להזין את שלושת הנתונים, ובסיום הפעולה תודיע לו על תוצאות החישוב.

```
10 PRINT 'THIS IS A TEST PROGRAM'
20 PRINT 'ENTER 3 NUMBERS'
30 INPUT A,B,C
40 LET X=(A+B+C) / 3
50 PRINT "THE AVERAGE IS:"; X
60 R = A / (B*C)
70 PRINT "THE RESULT OF A / (B*C) IS:"; R
80 END
```

נסקור עתה את הוראות התכנית:

10 הדפסת שורה, אשר לצורך זה תהיה כותרת לסדרת הפעולות והדיווח שבהמשך.

20 הודעה למשתמש להזין שלושה נתונים.

30 חישוב הממוצע. שים לב, שסוגריים חובקים את סכום המספרים, כדי שפעולה זו תתבצע ראשונה, ורק אחריה יבוצע החילוק. השימוש במלה LET אינו הכרחי.

50 הדפסת התוצאה. הערך שחושב יודפס בהמשך השורה לאחר דברי ההסבר.

60 חישוב מנת החילוק. שים לב, שסוגריים חובקים את המכפלה, כדי שתתבצע תחילה, לפני פעולת החילוק.

70 הדפסת התוצאה של החילוק. הערך שחושב יודפס בהמשך של שורת ההסבר.

80 הוראת סיום התכנית.

בעת ביצוע התכנית יוצגו על המסך (או יודפסו במסוף מכונת כתיבה):

```
THIS IS A TEST PROGRAM
ENTER 3 NUMBERS
?
12, 2, 4
```

נניח שאלו המספרים שיוזנו:

```
THE AVERAGE IS:6
THE RESULT OF A/(B*C) IS:1.5
```

14.7 הוראות בקרה

הביצוע הרגיל של הוראות התכנית הינו לפי סדר כתיבתן בזו אחר זו. לעתים יש צורך לחרוג מסדר רצוף זה, כדי לבצע הוראות אחרות או להפסיק את ביצוע התכנית. שינוי המסלול של ביצוע התכנית נעשה על ידי <u>הוראות בקרה</u> (CONTROL STATEMENTS), אשר כוללות הוראות שונות: פקודות דילוג, משפטי תנאי, פקודות עצירה, ופקודות המשך ועוד. לדוגמה, הפיתרון שהוצג קודם לכן למשוואה הריבועית, לא היה שלם, מכיון שלא נבדק, אם הביטוי לחישוב השורש הוא חיובי כדי שאפשר יהיה לחשב אותו. אפשר להטיל על המחשב את ביצוע הבדיקה ולפעול בהתאם לתוצאה. כלומר לעבור לקטע תכנית, אשר יבצע את החישוב המתאים, או שיפסיק את התכנית.

14.7.1 הוראת דילוג - GOTO

השימוש בפקודת דילוג יודגם ע"י הבעייה הבאה. נניח שדרושה הדפסה של המספרים מ-1 עד 10 והריבוע של מספרים אלה, במבנה

```
 1    1
 2    4
 3    9
---  ---
10   100
```

נשתמש לשם כך במשפטי הדפסה:

שורת התכנית הראשונה:

```
510 PRINT   1,1**2
520 PRINT   2,2**2
530 PRINT   3,3**2
```

שורת התכנית האחרונה:

```
600 PRINT 10,10**2
610 END
```

אם תידרש הדפסה של סדרת מספרים ארוכה יותר, נצטרך לכתוב שורות רבות של פקודות, דבר שאינו מעשי. נשתמש עתה בהוראת הדפסה אחת בלבד, אשר נשנה בה מדי פעם את הנתון המודפס. כלומר נשתמש במשתנה במקום בקבוע. כדי לחזור לפקודה, יש לדלג אל ההוראה המתאימה, כלומר להשתמש בהוראת GOTO או GO TO (עם רווח). התכנית תיראה כך:

```
10    M=1
20    PRINT M, M**2
30    LET M = M + 1
40    GOTO 20
50    END
```

ההוראה GOTO שבשורה 40 גרמה לדילוג אל השורה, שמספרה מופיע בשורה GOTO, זוהי שורה 20. לאחר מכן יימשך ביצוע התכנית לפי סדר השורות או ההוראות.

הסבר התרגיל:

10 קביעת ערך התחלתי למשתנה (במקרה שלנו=1)

20 הדפסה. M=1 ולכן יודפס: 1 1.

30 הגדלת ערכו של המשתנה ("מונה") ביחידה אחת, כעת M=2.

40 פקודת דילוג. חוזרים לשורה 20 לביצוע הדפסה, אולם כעת M=2 וההדפסה תהיה: 2 4. (אילו חזרנו לשורה 10 היינו מקבלים מצב התחלתי שבו M=1, וזו לא כוונתנו).

30 הגדלת ערך המשתנה ביחידה אחת. כעת M=3.

40 הוראת דילוג לשורה 20.

20 הוראת הדפסה כאשר M=3 תגרום להדפסה 3 9 וכן הלאה וכן הלאה, <u>עד אין סוף</u>.

כיצד עוצרים, לאחר שהודפס מצב שבו M=10? נעשה זאת באמצעות <u>הוראת התנייה</u>, אשר נכיר מיד.

במספר מערכות אפשר לרשום הוראת דילוג מורכבת יותר, אשר מבצעת דילוג לפי ערך של משתנה או לפי תוצאת חישוב של ביטוי המופיע בפקודה.

לדוגמה:

```
100   GOTO 80, 40, 60 ON K
200   GOTO 80, 40, 60 ON (A+B)/2
```

בשני המקרים יבוצע סיעוף לשורה 80, כאשר ערך המשתנה או ערך הביטוי הוא 1, דילוג לשורה 40, כאשר הערך הוא 2, דילוג לשורה 60, כאשר הערך הוא 3. אם הערך <u>שונה</u> מ-1, 2, 3 (מכיון שיש רק 3 אפשרויות לביצוע הדילוג) תמשיך התכנית בשורה העוקבת להוראה GOTO.

במערכות מסויימות תירשם ההוראה שבשורה 100 בצורה הבאה:

```
100 ON K GOTO 80, 40, 60
```

14.7.2 הוראות התניה - IF

הוראת התניה מורה למחשב לבדוק ערכו של משתנה או של ביטוי ע"י השוואה לגורם קבוע או למשתנה אחד, ולדלג לשורה כלשהי בתכנית או לבצע פעולה כלשהי. קיימות צורות אחדות להוראת ההתניה. ההוראה הבסיסית היא "אם", או IF STATEMENT.

נחזור לדוגמת ההדפסה של סדרת המספרים. רוצים לעצור לאחר הדפסת המספר 10 והריבוע שלו, ולא להמשיך בדילוגים לשורה 20. נכתוב זאת כך:

```
10    LET M=1
20    PRINT M,M**2
30    M = M+1
40    IF M <= 10 GOTO 20
50    END
```

ההוראה בשורה 40 עורכת את ההשוואה של M ל-10. אם ערכו של M קטן או שווה ל-10, יש להמשיך בהדפסה ובהגדלת ערכו של M, ואם לא, כלומר M גדול מ-10 (המצב הוא M=11), יש לעבור להוראה 50, לסוף התכנית.

<u>שים לב</u>, אם הוספת הערך למשתנה ואם הבדיקה שלו נעשים <u>לפני</u> ביצוע הפעולה או <u>אחריה,</u> כי בכך תלויה ההחלטה על דילוג או על המשך העבודה.

בכמה מערכות מבנה הפקודה IF שונה, ולאחר הצגת התנאי יש לכתוב THEN, ואין זה שונה מהמצב הקודם עם GOTO, למשל, M<=10 THEN 20 IF 40.

הוראת ההתניה בתרגיל הקודם קרויה גם <u>הוראת התניה לוגית</u> (LOGICAL IF STATEMENT). היא כוללת ביטוי לוגי והתשובה, אשר ניתנת בסיום הבדיקה, היא האם הוא <u>נכון</u> (TRUE) או <u>לא נכון</u> (FALSE), או במלים אחרות האם היחס הלוגי המוצג <u>מתקיים</u> או <u>לא מתקיים.</u> הצורה הכללית של ההוראה היא:

IF < ביטוי לוגי > ביטוי

הביטוי הלוגי מבטא <u>יחס בין שני גורמים</u>, אשר יכולים להיות קבועים, משתנים או ביטויים. היחס יכול להיות שווה, גדול, קטן וכדי. הביטוי מימין הוא הוראה, שיש לבצע, אם הביטוי הלוגי הוא <u>נכון,</u> כלומר אם התנאי של הביטוי <u>מתקיים.</u> לאחר ביצוע הביטוי תמשיך התכנית בשורה העוקבת. אם הביטוי הוא הוראת דילוג, תמשיך התכנית בשורה, שרשומה בהוראת הדילוג. אם הביטוי

הלוגי <u>אינו נכון</u>, כלומר התנאי <u>אינו מתמלא</u>, תמשיך התכנית בשורה העוקבת. כאשר "הביטוי" הוא הוראת דילוג, אפשר לכתוב אותו כך:

IF < ביטוי לוגי > GOTO 25
IF < ביטוי לוגי > THEN 25

קיימות מספר אפשרויות לכתיבת ביטויים לוגיים וצורות כתיבה אחדות לכתיבת אותו ביטוי, בהתאם להצגה של <u>סימן היחס</u> (RELATIONAL OPERATOR). האפשרויות השונות ניתנות בטבלה, שבה A ו-B מייצגים שני גורמים שונים, שיכולים להיות קבועים, משתנים או ביטויים אריתמטיים.

היחס	אפשרויות להצגת היחס		
A שווה ל-B	EQUAL	A .EQ. B	A=B
A אינו שווה ל-B	NOT EQUAL	A .NE. B	A#B, A<>B
A גדול מ-B	GREATER THAN	A .GT. B	A>B
A קטן מ-B	LESS THAN	A .LT. B	A<B
A גדול או שווה ל-B	GREATER OR EQUAL	A .GE. B	A⩾B, A>=B
A קטן או שווה ל-B	LESS OR EQUAL	A .LT. B	A⩽B, A<=B

בכמה מערכות אפשר להשתמש רק בחלק מהאפשרויות של הצגת היחס ובדרך כלל בצורה הקצרה. ישנן מערכות, שבהן אין צורך לרשום נקודות משני צדי סימן היחס (למשל EQ ולא .EQ.), אולם אם רושמים נקודות, הן חייבות להיות צמודות לסימן היחס. ביניהן לבין הגורמים יכול להיות רווח. אפשר ואף רצוי להשאיר רווחים משני צדי סימן היחס, אך אם סימן היחס מורכב משני סימנים, כמו =< הם חייבים להיות צמודים.

<u>14.7.3 הוראת עצירה - STOP</u>

כדי לעצור את ביצוע התכנית, יש להשתמש בהוראה STOP. כשמגיעה התכנית אל נקודה זו, מופסק הביצוע, והפיקוח עובר אל תכנית הניהול של המחשב, אל מערכת ההפעלה. כאשר סוף הביצוע של התכנית הוא בפקודה האחרונה שלה, במקום שנמצאת ההוראה END, אפשר לוותר על כתיבת STOP, כי END משמשת גם למטרה זו. כאשר משתמשים בהוראות התנייה ודילוג, יכול להיווצר מצב, שבו סוף הביצוע של התכנית אינו בהוראה האחרונה שלה.

<u>תרגיל</u>

יש לקרוא מספר ערכים, אשר אינו קבוע מראש, לחשב את הממוצע ולמצוא את המספר הגדול שבהם. המחשב ידע כמה ערכים לקבל לפי המספר הראשון, שיוזן לו.

`10 M=0`	מונה הסכום (מצב התחלתי = 0)
`20 K=0`	המספר הגדול (מצב התחלתי = 0)
`30 INPUT "HOW MANY NUMBERS:"; N`	מספר הערכים שיוזנו
`35 A=N`	שמור את N
`40 INPUT "NEXT NUMBER TO SUM";`	הזנת ערך נוסף
`50 M = M+X`	הוסף לסכום
`60 IF X>K THEN 130`	בדוק אם X גדול מ-K
`70 N = N-1`	הורד 1 מהמונה ובדוק:
`80 IF N#0 THEN 40`	אם N#0, המשך לקרוא
`90 B = M/A`	חישוב הממוצע
`100 PRINT "AVERAGE OF"; A; "NUMBERS IS:" ;B`	
`110 PRINT "THE LARGEST NUMBER IS:"; K`	
`120 STOP`	הפסקת התכנית
`130 K=X`	העבר את X לתא של המספר הגדול
`140 GOTO 70`	חזור להמשך התכנית
`150 END`	סוף התכנית

ואיך תהיה ההדפסה של התוצאה? לדוגמה,

בשורה הראשונה: `AVERAGE OF 7 NUMBERS IS: 53.5`

בשורה השנייה: `THE LARGEST NUMBER IS: 138`

עתה נחזור להוראות ההתניה, ונכיר הוראות התניה מורכבות יותר.

14.7.4 הוראות התניה מורכבות

ההוראה הפשוטה כללה יחס בין שני משתנים. עתה נכיר צורות נוספות של יחסים ושל הוראות סיעוף מתוך הוראת ההתניה. נציג מספר דוגמאות והסבר להן:

```
30   IF A = 20 GOTO 140
35   IF A = 20 THEN 140
40   IF A = 20 THEN 140 ELSE 250
50   IF A * 2 < 35/B GOTO 180
60   IF A > 5 THEN PRINT A ELSE 80
```

35,30 הוראת סיעוף פשוטה. כאשר התשובה לשאלה היא <u>כן</u>, כלומר כאשר A=20, יבוצע הדילוג. אם לא - תתבצע השורה העוקבת. שים לב, לכך שבכמה מערכות נוהגים לכתוב THEN במקום GOTO כפי, שמוצג בשורה 35, אך אלו הן הוראות זהות.

40 בשורה זו מתבצעת בדיקה, כפי שנעשתה בשורה 35. כאשר התנאי של הבדיקה יתמלא, כלומר כאשר A=20, יתבצע דילוג לשורה 140. אם התשובה היא <u>לא</u>, יתבצע הדילוג לשורה, שמספרה מופיע לאחר המלה ELSE, כלומר

לשורה 250 (ולא לשורה העוקבת כמו בדוגמה הקודמת). במקרה זה אין מעבר בין ההוראה שבשורה 40, להוראה שבשורה העוקבת. התכנית תגיע לשורה העוקבת לשורה 40 ע"י הוראת דילוג ממקום אחר.

50 הוראה זו אינה שונה מההוראה בשורה 30. כאן ישנם ביטויים במקום ערכים קבועים או משתנים בודדים. בתחילה יחושבו הביטויים, והבדיקה בהוראה IF תבוצע על תוצאות החישוב. כלומר אם A * 2 קטן מ-35/B יבוצע דילוג, ואם התשובה שלילית, תמשיך התכנית לשורה העוקבת.

60 בהוראה זו מוצגת אפשירות נוספת של התניה. אם התנאי מתמלא, אפשר לבצע הוראות שונות ולאו דווקא הוראת דילוג. לדוגמה אם התנאי מתמלא, תבוצע הדפסה של ערך המשתנה A, ולאחריה תמשיך התכנית בשורה העוקבת. אם התנאי לא מתמלא, יבוצע סיעוף לשורה 80.

הצגנו קודם דוגמאות של הוראות התניה עם קבועים, משתנים וביטויים מספריים. אפשר לכלול בהוראת IF גם משתנים אלפנומריים ומחרוזות של תווים. ההשוואה תבוצע על פי סדר המיון של התווים. למשל:

אם נכתוב ......'AD' < 'AB' IF התשובה היא <u>כן</u>, מכיון ש-B מופיע לפני D בסדר המיון. בהוראה אחרת .....IF X$ = Y$ תושווינה מחרוזות התווים במשתנים אלו זו לזו. אם יהיה שוויון, נקבל תשובה <u>כן.</u>

<u>14.8 הוראת הערה - REM</u>

הוראת REM (קיצור של REMARK או REMINDER) אינה הוראת ביצוע אלא הוראה הערה או תזכורת, אשר נוספת לתכנית המודפסת על ידי המתכנת לשם תזכורת והסבר. תמליל זה אינו נמצא בזיכרון המחשב, ואינו תופש בו מקום. המקום היחיד הנדרש הוא באמצעי האחסון של התכנית (סרט או דיסק). כאשר מבקש המתכנת להציג את התכנית לעיון, היא נכנסת לזיכרון בשלימותה. שמת לב בודאי לעובדה, שהאותיות של המשתנים בלבד אינן מסייעות להבנת התכנית, אפילו היינו כותבים שמות משתנים בני אותיות אחדות במקום שמות בני אות אחת.

הצורה של הוראה REM היא:

```
15    REM THIS IS A MESSAGE
```

המלה REM מורה למחשב, שכל מה שמופיע אחריה אינו שייך להוראות

התכנית, עד להוראה הבאה. אפשר לכתוב את ההוראה REM בהמשך של הוראות אחרות, למשל:

```
20    A=A+5  REM ADD 5 TO COUNTER
```

<u>תרגיל</u>

נחזור עתה לפתרון המשוואה הריבועית, ונשלים את התכנית ע"י הוספת בדיקה של הביטוי B-4AC. התכנית המשופרת תיראה כך:

```
 20    INPUT "ENTER 3 NUMBERS:"; A,B,C
 30    D = (B**2 - 4*A*C)
 40    IF D < 0 THEN 100
 50    REM CALCULATE X1 AND X2
 60    K = D**.5
 70    X1 = (-B+K)/(2*A)
 80    X2 = (-B-K)/(2*A)
 85    PRINT "X1=";X1;"  X2=";X2
 90    STOP
100    PRINT "IMPOSSIBLE TO CALCULATE SQRT OF D."
110    GOTO 90
120    END
```

<u>הסבר</u>

20 — בתחילת התכנית יש הודעה להזין 3 מספרים, אשר יהיו שלושת המקדמים של המשוואה.

30 — חישוב "דלתה", הביטוי שתחת סימן השורש במשוואה הריבועית.

40 — בדיקה אם הערך של דלתה קטן מאפס. במקרה זה אי-אפשר לחשב שורש ועוברים לשורה 100 להצגת הודעה ולסיום התכנית.

50 — במקרה שהערך של דלתה אינו קטן מאפס זוהי שורת הערה (REM), שאינה פקודת ביצוע, ואין לה כל משמעות מבחינת התכנית, היא מיועדת למתכנת, אשר קורא את התכנית.

60 — לאחר ביצוע הפקודה בשורה 40, תעבור התכנית לשורה זו לחישוב השורש של דלתה. אם D=0, גם K=0 ויכולנו לחסוך את ביצוע ההוראה, אך אין לזה חשיבות במקרה זה.

80,70 — חישוב שורשי המשוואה הריבועית X1 ו-X2.

85 — הצגת שני השורשים של המשוואה בשורה אחת.

90 הוראת סוף הביצוע של התכנית. מכאן תעבור התכנית למערכת ההפעלה.

100 לכאן תגיע התכנית מההוראה שבשורה 40, כאשר דלתה קטן מאפס.

110 כאן אפשר לסיים את התכנית, אך מקובל לשמור על <u>הוראת סיום אחת בלבד בתכנית.</u> ייתכן שלפני סיום התכנית נרצה להציג הודעה כלשהי או לבצע הוראות סיום (למשל בעת שעובדים בקבצים, בסרטים ובדיסקים) ועל כן עדיף שהסיום יהיה במקום אחד בלבד.

120 פקודה END מציינת את הסוף הפיזי של קבוצת הפקודות של התכנית.

אם המשוואה, שצריכים לפתור, היא $X^2-7X+10$, המקדמים שלה בהתאמה הם 1, 7, 10 והתוצאה המודפסת בגמר החישוב היא: X2=5, X1=2.

<u>תרגיל</u>

אברהם מגיע לחנות לקנות 3 פריטים, ובידו שטר של כסף. לאחר הקנייה הוא ניגש לקופאי, אשר מסכם את המחירים של הפריטים שנרכשו. אם הסכום שווה לכמות הכסף שבידו, אין צורך בפעולה נוספת, ואברהם עוזב את החנות. אם סכום הקנייה נמוך יותר, יש לחשב את סכום העודף. אם סכום הקנייה גבוה, על אברהם להחזיר את הפריט האחרון שלקח (בדוגמה שלנו - C) וחוזרים על הבדיקה. אם גם עתה הסכום גבוה, מחזיר אברהם את הפריט שלפניו (B), וחוזרים על הבדיקה. אם המחיר של A בלבד גבוה מדי ואין בידו לשלם, עליו לבחור פריטים חדשים, ולמעשה לחזור על כל התהליך מתחילתו.

את התהליך, שהצגנו, אפשר להציג בתכנית מחשב:

אברהם יקיש תחילה את סכום הכסף שבידו ואת המחירים של הפריטים שקנה. ואת המחיר "0" במקום המחירים של פריטים, שלא קנה. התכנית פשטנית למדי, כי מטרתה להדגים שימוש בפקודות שפת בייסיק ולא להציג פתרון לבעייה מסוימת. במציאות יבחר אברהם מהו הפריט שיחזיר כדי להישאר בתחום תקציבו, ולא יחזיר פריטים לפי סדר כלשהו קבוע.

```
10   PRINT 'PURCHASE PROGRAM'
15   INPUT 'YOUR MONEY'; T
20   INPUT 'PRICES OF 3 ITEMS'; A,B,C
30   LET S=A+B+C            REM CALCULATE AMOUNT TO PAY
40   IF S=T GOTO 151        REM EXACT AMOUNT
50   IF S<T GOTO 140        REM CHANGE REQUIRED
60   IF C=0 GOTO 90         REM ITEM C NOT PURCHASED OR
                                                RETURNED
```

```
65   PRINT 'RETURN ITEM C'
70   C=0
80   GOTO 30                  REM START CALCULATIONS
90   IF B=0 GOTO 120  REM ITEMS B, C NOT PURCHASED OR
                                              RETURNED
95   PRINT 'RETURN ITEM B'
100  B=0
110  GOTO 30
120  PRINT 'PLEASE CHOOSE OTHER 3 ITEMS'
130  GOTO 20                  REM START FROM INPUT PHASE
140  R=T-S REM CALCULATE CHANGE
150  PRINT "YOUR CHANGE IS:";R;' SHEKEL'
151  PRINT "YOUR PURCHASE IS:";S;' SHEKEL'
160  END
```

<u>הסבר</u>

10 שם התכנית, יוצג בתחילת העבודה.

15 קלט של סכום הכסף (T) שברשות הקונה.

20 קלט מחירי 3 מוצרים (C,B,A).

30 סיכום מחירי הפריטים.

40 בדיקה אם סכום הקנייה שווה ל-T. אם <u>כן</u>, עוברים לשורה 151 להצגת סכום הרכישה ולסיום התכנית.

50 בודקים, אם סכום הרכישה קטן מ-T. אם כך הוא, אז זכאי אברהם לעודף, אשר יחושב בשורה 140. בשורה 150 יוצג סכום העודף, במשפט 151 יוצג סכום הקנייה והתכנית תסתיים.

60 התכנית תגיע לשורה זו, כאשר התשובה בהוראות ההתניה שבשורות 40 ו-50 היתה <u>לא</u>, כלומר סכום הקנייה היה גבוה מ-T. אם לא קנה אברהם פריט C, או שכבר החזיר אותו, עוברים להוראה 90, ואם לא עוברים לשורה העוקבת.

65 אברהם מתבקש להחזיר את פריט C.

70 לאחר שהחזיר, ערכו של משתנה C יהיה "0" (אפס).

80 עוברים לשורה 30, לתחילת החישוב של סכום הקנייה. עכשיו, של פריטים A ו-B בלבד (C=0).

90 אם אברהם לא קנה או לא החזיר פריט C מגיעים לשורה זו. בודקים, אם לא קנה או כבר החזיר פריט B. אם <u>כן</u> - עוברים לשורה 120. אם לא - ממשיכים.

95 אברהם מתבקש להחזיר פריט B.

100 ערכו של B יהיה "0" (אפס).

110 עוברים לשורה 30, לתחילת החישוב, שבו רק A#0.

120 לשורה זו מגיעים, אם פריטים B ו-C לא נקנו, או שכבר הוחזרו, ומחירו של A גבוה מ-T. אברהם מתבקש לבחור 3 פריטים חדשים.

130 חוזרים לתחילת התכנית.

140 חישוב העודף.

150 הצגת הודעה שכוללת את סכום העודף שחושב.

151 הצגת הודעה ובה ערך הקנייה.

160 פקודה לסיום התכנית.

14.9 אופרטורים לוגיים

אפשר לבנות ביטוי לוגי מצירוף של ביטויים, אשר ממלאים תנאים שונים, והבדיקה בהוראה IF תתייחס לביטוי הלוגי המורכב. צירוף של ביטויים יכול להיעשות באמצעות אחת מהוראות הפעולה הבאות:

הסבר	אפשרות להצגת התנאי	
לא	¬	.NOT.
וגם	&	.AND.
או	|	.OR.

הערה: בכמה מערכות אין צורך לרשום את הנקודות שמשני צדי המלים.
העדיפות לביצוע ההוראות היא לפי סדר כתיבתן.

דוגמה: אם A שווה ל-B וגם C גדול מ-D יש לבצע דילוג. אם לא, ממשיכים בשורה העוקבת. כלומר הדילוג יבוצע, רק אם שני התנאים מתמלאים ("וגם").

```
80   IF A=B AND C>D GOTO 110
```

בדוגמה הבאה נבדוק, אם הערך המספרי של משתנה B קטן מהערך

המספרי של משתנה A קטן מ-B, או הערך של משתנה אלפנומרי C$ שווה ל-"DOG". הדילוג יבוצע, כאשר לפחות תנאי אחד מתמלא ("או").

```
60   IF A<B OR C$ = "DOG" GOTO 170
```

תרגיל

המורה להתעמלות החליט להעניק פרס ראשון לתלמידים שצברו לפחות 90 נקודות בבחינה בתנ"ך (כלומר הציון הוא: "גדול או שווה" ל-90) וגם לפחות 85 נקודות בבחינה בספרות. פרס נחומים יינתן לתלמידים, שצברו לפחות 95 נקודות בתנ"ך או לפחות 90 נקודות בעברית, או לפחות 80 נקודות באנגלית. הקלט לתכנית יהיה מורכב משם התלמיד ו-3 מספרים בעבור כל אחד מהמקצועות (B - תנ"ך, H - עברית, E - אנגלית). אם מקישים 0 במקום הציון בתנ"ך, תבין התכנית, שזהו סוף הקלט.

```
 20   INPUT "NAME AND 3 NUMBERS"; NAME$, B, H, E
 30   IF B=0 GOTO 100
 40   IF B>=90 AND H>=85 GOTO 80
 50   IF B<95 OR H<90 OR E<80 GOTO 20
 60   PRINT "PRIZE 2 TO:"; NAME
 70   GOTO 20
 80   PRINT "PRIZE 1 TO:"; NAME
 90   GOTO 20
100   END
```

הסבר

20 קריאת הקלט: שם התלמיד ו-3 ציונים.

30 בדיקה אם B=0. אם כן, עוברים לסוף התכנית בשורה 100.

40 הוראת התניה עם אופרטור לוגי AND. אם A גדול או שווה ל-90 וגם B גדול או שווה ל-85 מדלגים לשורה 80 להדפסה, פרס 1.

50 הוראת התניה עם אופרטור לוגי OR.
שים לב, שבשורה זו "הפכנו" את סדר הבדיקה, ובמקום לבדוק, אם B גדול או שווה ל-95, בדקנו, אם B קטן מ-95 (וכך גם בעבור H ו-E). הדבר נעשה מפאת נוחות התכנית וכיוון הדילוגים.
אם B קטן מ-95 או H קטן מ-90 או E קטן מ-80, אזי לא יקבל התלמיד כל פרס, ומדלגים לשורה 20 לקליטת נתונים של תלמיד אחר. אם התנאי לא מתמלא, כלומר התלמיד צבר מספר נקודות מינימלי לפחות במקצוע אחד, הוא מקבל פרס שני, שורה 60.

70,60 הדפסת שם התלמיד שקיבל פרס שני ודילוג להתחלה.

90,80 הדפסת שם התלמיד שקיבל פרס ראשון ודילוג להתחלה.

100 סוף התכנית (סוף ביצוע וגם סוף רשימת ההוראות).

14.10 לולאות

לעתים יש צורך לבצע סדרת הוראות באופן מחזורי, בסיכום של סדרת מספרים למשל, יש לחזור על פעולת החיבור בעבור כל אחד מהמספרים עד לסיום הסדרה. כאשר יש לחשב ממוצע ציונים של תלמידים, יש לקרוא את רשומות הנתונים של כל התלמידים, כאשר בכל רשומה נמצאים נתוני הציונים של כל תלמיד, ועל פיהם מחושב הממוצע. פעולות הקריאה, החישוב וההדפסה של התוצאה תחזורנה עד לסיום קובץ הנתונים, וכבר נתקלנו בבעייה דומה בתרגיל קודם.

נלמד עתה, כיצד לכתוב תכנית, אשר נדרשת בה חזרה על סדרת פקודות. הפקודות עשויות להיות קבועות ואחידות, או שניתן להתנות את ביצוע כולן או חלקן בנתונים משתנים המתקבלים בעת ביצוע התכנית.

<u>הלולאה</u> (LOOP) הינה חזרה על סדרת פקודות במחזוריות. הבקרה של הלולאה, או מספר הפעמים, שחוזרים על סדרת הפקודות, נקבעים על ידי אינדקס או תנאי כלשהו. האינדקס הוא משתנה מספרי.

<u>תרגיל</u>

נתונה סדרת מספרים (DATA) יש לסכם 5 מהם ולהדפיס את התוצאה.

```
 10   REM SUM OF 5 NUMBERS
 20   DATA 15,5,2,14,8,9,30     REM LIST OF 7 NUMBERS
 30   T=0                       REM INITIAL CONDITIONS
 40   I=0                       REM I IS THE LOOP INDEX
 50   READ X                    REM READ NEXT NUMBER
 60   T=T+X                     REM ADD TO COUNTER
 70   I=I+1                     REM ADD TO LOOP INDEX
 80   IF I<=5 GOTO 50           REM CHECK CONDITIONS
 90   PRINT "TOTAL=";T
100   END
```

<u>הסבר</u>

לאחר הקביעה של תנאי ההתחלה, קוראים מספר מתוך הוראת DATA ומסכמים אותו למונה T, מוסיפים 1 לאינדקס I (אשר אופס בתחילה)

ובודקים אותו (שורה 80). אם I קטן או שווה ל-5, המשמעות היא, שטרם סיימנו את חיבור 5 המספרים, כפי שהתכוונו לעשות, ולכן עוברים לשורה 50 להמשך. אם I גדול מ-5, כלומר התשובה להוראת ההתניה היא "לא", עוברים להדפסת התוצאה ולסיום התכנית.

אפשר לראות בדוגמה זו, ששלוש ההוראות, המרכיבות את הלולאה, הן:

- קביעת תנאי התחלה של האינדקס . `40 I=0`
- שיעור הקידום של האינדקס `70 I=I+1`
- בדיקת תנאי הסיום `80 IF I<=5 GOTO 50`

<u>שים לב</u>, שאפשר לקבוע תנאי התחלה שונים, ולהציב את ההוראה הקובעת את שיעור הקידום, I=I+1, למשל, לפני או אחרי הוראת הבדיקה. במקרה זה סימן היחס בהוראת IF יהיה שונה. דוגמה לכך בתרגיל הבא:

<u>תרגיל</u>

יש לחשב את הסכום של המספרים מ-20 עד 40.

```
10    REM SUM OF NUMBERS FROM 20 TO 40
15    M=20                 REM INITIAL CONDITIONS FOR INDEX M
20    S=0
25    S=S+M                  REM CALCULATE SUM
30    IF M>=40 GOTO 45       REM EXIT CONDITIONS
35    M=M+1                  REM INCREASE INDEX
40    GOTO 25
45    PRINT "SUM=";S
50    END
```

הדפסת התוצאה בגמר התכנית: SUM = 630

<u>הסבר</u>

לאחר הקביעה של תנאי ההתחלה, המספר הראשון, שנסכם לתוך מונה S הוא 20 (M=20). נבדוק את הערך M. אם הוא "גדול או שווה" ל-40, נעבור לסיום התכנית. אם לא - כלומר M<40, נוסיף 1 לאינדקס ונמשיך לחבר למונה S. נבדוק שוב וכן הלאה, עד שהתשובה בהוראה IF תהיה "כן", ונעבור לשורה 45. שים לב, שבדוגמה זו האינדקס M משרת שתי מטרות, האחת - גורם שאנו משתמשים בו לחיבור למונה, והשנייה - אינדקס לולאה. דבר נוסף, שיש לשים אליו לב, הוא, שתנאי ההתחלה של M הוא המספר הראשון, שאנו רוצים לסכם ולא אפס כמו בדוגמה הקודמת. מלבד זאת, הוראת הקידום של האינדקס נמצאת לאחר הבדיקה ולא לפניה, ולכן סימן היחס בהוראה IF שונה מאשר בדוגמה הקודמת.

<u>14.10.1 הוראה לבקרת לולאה - FOR-NEXT</u>

מכיון שהשימוש בלולאות נפוץ ביותר, נקבעו הוראות מיוחדות

למטרה זו. ההוראות מאפשרות למתכנת לבצע סדרה של פעולות חוזרות בלולאה ולפקח על ביצוע זה עד לסיומו. הפיקוח על ביצוע הלולאה מתאפשר על ידי שימוש באינדקס, אשר משתנה בין שני ערכים, אשר יכולים להיות קבועים או משתנים.

ההוראות הבסיסיות הן :

```
40   FOR   I=B TO K
-- -- -- -- --
     הוראות
-- -- -- -- --
70   NEXT I
```

ההוראה FOR קובעת את תחום הערכים לאינדקס I: הערך התחתון הוא B (מספר קבוע, משתנה או ביטוי) והערך העליון הוא K; השינוי של האינדקס, אם לא נאמר אחרת, הוא 1 (זוהי "ברירת מחדל"). תחום הלולאה הן השורות עד שורה 70. כאשר האינדקס I קטן מ-K, תעבור התכנית לשורה שלאחר הוראת FOR ותבצע את הלולאה פעם נוספת. כאשר האינדקס שווה או גדול מ-K, תמשיך התכנית בשורה העוקבת לשורה 70 אשר מכילה את ההוראה NEXT. שים לב, לכך שהערך של האינדקס בסיום הביצוע של הלולאה יכול להיות K או K+1. הדבר תלוי בשיטת העבודה של בייסיק במחשב שבו מופעלת התכנית.

נשתמש עתה בהוראות הלולאה, כדי לחשב את הסכום של המספרים מ-20 עד 40. על הלולאה נשלוט בעזרת משתנה M, המתקדם מ-20 עד 40 בדילוג של יחידת אחת.

```
10   S=0
20   FOR M=20 TO 40
30   S=S+M
40   NEXT M
50   PRINT "SUM=";S
60   END
```

10 איפוס המונה.

20 פקודה, המציינת ערך התחלתי M=20 ושינוי M בקידום של 1 בכל פעם שההוראה תתבצע, עד ל-M=40. כלומר החישוב יבוצע בעבור הערכים: M=20,21,22....39,40.

30 פקודת החיבור של ערך M למונה.

40 פקודה זו מציינת, שעד אשר M לא ממלא את התנאי של סיום הלולאה, כמפורט בשורה 20, יוחזר הפיקוח לשורה 20 (שבה צוין ...FOR M).

50 כאשר M יהיה גדול מ-40, תופסק הלולאה, והפיקוח יועבר לשורה שלאחר פקודת NEXT. התוצאה תוצג והתכנית תסתיים.

בין פקודת FOR לפקודת NEXT יכול להיות כל מספר של פקודות ואפשר לכלול בהן גם לולאות נוספות.
דוגמה נוספת ללולאה, שבה תודפס חזקה שלישית של המספרים 101 עד 105 (כולל):

```
20   FOR N=101 TO 105
30   PRINT N**3
40   NEXT
```

אין צורך לרשום את שם המשתנה בפקודת NEXT, אם היא מתייחסת לפקודת FOR אחת.

14.10.2 קידום לולאה בדילוג - FOR-STEP

בדוגמאות הקודמות, התקדם אינדקס הלולאה ב-1. לעתים יש צורך לקדם את האינדקס בערך שונה מ-1, למשל, ב-2. כאשר רוצים לסכם את המספרים הזוגיים מ-14 עד 28, למשל, דרוש קידום בזוגות ולא ביחידות. הפקודה FOR תהיה:

```
60   FOR M=14 TO 28 STEP 2
     -- -- --
90   NEXT M
```

הביטוי STEP 2 מציין את שיעור הקידום של האינדקס M.

14.10.3 לולאות מורכבות

את הוראת הלולאה FOR אפשר לכתוב בצורות שונות, כאשר במקום הקבועים שבדוגמה (14, 28, 2 בדוגמה האחרונה) אפשר לרשום משתנים או ביטויים אריתמטיים:

```
20   FOR I=A TO 70 STEP W
60   FOR K=8*R TO B-C STEP 2
90   FOR Z=5 TO R+2 STEP A/B
```

20 הערך ההתחלתי של I נלקח ממשתנה A והלולאה תבוצע, עד אשר I יהיה גדול מ-70. קידום הערך I בכל ביצוע של הלולאה יהיה לפי הערך של W.

60 בהוראת הלולאה מופיעים ביטויים חשבוניים. הערך ההתחלתי הוא 8*R והערך הסופי B-C. קידום אינדקס הלולאה ב-2.

90 הערך הגבוה של הלולאה הוא ביטוי R+2, והקידום של האינדקס Z הוא גם כן ביטוי, שערכו A/B.

הביצוע של הלולאות יופסק, כאשר הערך של האינדקסים I, K או Z יהיה שווה או גבוה מהערכים שלאחר המלה TO בהוראות FOR השונות. שים לב, שמשך ביצוע של לולאה, המכילה ביטויים אריתמטיים, הוא <u>ארוך יותר</u> ממשך הביצוע של לולאה, שבה יש מספרים קבועים או משתנים בלבד.

<u>14.11 לולאות מרובות</u> או <u>לולאות מקוננות</u>

תהליכי עיבוד מסויימים מחייבים ביצוע <u>לולאות מקוננות</u> (NESTED LOOPS), שבהן יש להכיל לולאה בתוך לולאה. בתוך קטע של תכנית המהווה לולאה אפשר להכיל לולאה, אחת או יותר, המקיפה חלק כלשהו של ההוראות. מקובל לכנות לולאות אלו <u>לולאה חיצונית</u> <u>ולולאה פנימית</u> בהתאמה.

אפשר לתאר זאת בצורה סכימטית בדוגמה, שבה לולאה ב' הפנימית מקוננת בתוך לולאה א' החיצונית.

```
-------------------  120   FOR B  = ...
|       לולאה א'          ..........
|                         ..הוראות.
|    ---------------  150   FOR H  = ...
|   |                     ..........
|   |   לולאה ב'          ..הוראות.
|   |                     ..........
|   |_______________  180   NEXT H    ...
|                         ..........
|                         ..הוראות.
|___________________  200   NEXT B    ...
```

קינון לולאות יכולה להיעשות בצורות שונות. את ראשית הלולאות אפשר לרשום במשפטים סמוכים, אם דרוש.

```
      -------------  120   FOR  B = ...
      |      -----  121   FOR  H = ...
      |      |            .........
 א'   |  ב'  |            .........
      |      |____  140   NEXT H
      |                   .........
      |___________  180   NEXT B
```

אפשר לרשום את סיום הלולאות במשפטים סמוכים או באותו משפט.

```
            ------------  120   FOR  B = ...
            |                   .........
            |       -----  140  FOR  H = ...
       א'   |  ב'  |            .........
            |      |_____  190  NEXT H       או     190 NEXT H,B
            ------------  191   NEXT B
```

בכל אחת מהדוגמאות, ראינו שהוראת NEXT לסיום הלולאה הפנימית נרשמת <u>לפני</u> הוראת NEXT של הלולאה החיצונית. אפשר לציין כלל זה בפשטות: יש לסיים את הלולאות בסדר הפוך מסדר הפתיחה שלהן. בדוגמה ראינו שתי לולאות בלבד, אך אפשר לקונן לולאות רבות יותר.

לדוגמה :

```
          -----------------  80    FOR  A = ...
          |                        .........
          |                        .........
          |    ------------  120   FOR  B = ...
          |    |      ------  121  FOR  C = ...
          |    |      |            .........
   א'     |  ב' |  ג'  |            .........
          |    |      |            .........
          |    |      |______  160  NEXT
          |    |                   .........
          |    |                   .........
          |___ |____________  200   NEXT A,B
```

בהוראה 160 לא רשמנו את שם אינדקס הלולאה, מדוע?

נדגים את השימוש <u>בלולאה כפולה:</u>

יש להדפיס את לוח הכפל למספרים מ-1 עד 10.

```
      ----------   10   FOR K = 1 TO 10
      |  -------   20   FOR M = 1 TO 10
      |  | M       30   PRINT K*M;
  K   |  |______   40   NEXT M
      |            50   PRINT
      |__________  60   NEXT K
                   70   END
```

מבנה הדפסה הרצוי הוא

```
        1    2    3    4    5    6    7    8    9   10
  10    2    4    6    8   10   12   14   16   18   20
שורות   3    6    9   12   .-   .-   .-   .-   .-   30
       --   --   --   --   --   --   --   --   --  ---
       10   20   30   40   50   60   70   80   90  100

                       10  עמודות
```

לולאות מקוננות מתבצעות בצורה הבאה: בתחילה מקבל המונה של הלולאה החיצונית K את הערך הראשון מההוראה FOR (במקרה שלנו - 1), והמונה של הלולאה הפנימית M מקבל גם הוא את הערך הראשון מההוראה FOR (במקרה שלפנינו - 1). אח"כ מתקדמת התכנית ומבצעת את ההוראות עד להוראה NEXT M, המציינת את סיום הלולאה הפנימית. המונה M יקודם ב-1 (אם לא ציינו אחרת בהוראה FOR שבשורה 20, פירוש הדבר, שהתכוונו לכתוב STEP 1, שמשמעותו קידום ב-1). אם המונה לא עבר את הערך העליון בהוראה FOR, יבוצע דילוג לתחילת הלולאה בהוראה 2. אם המונה עבר את התחום העליון שלו, תבוצע ההוראה הבאה שלאחר הוראת NEXT.

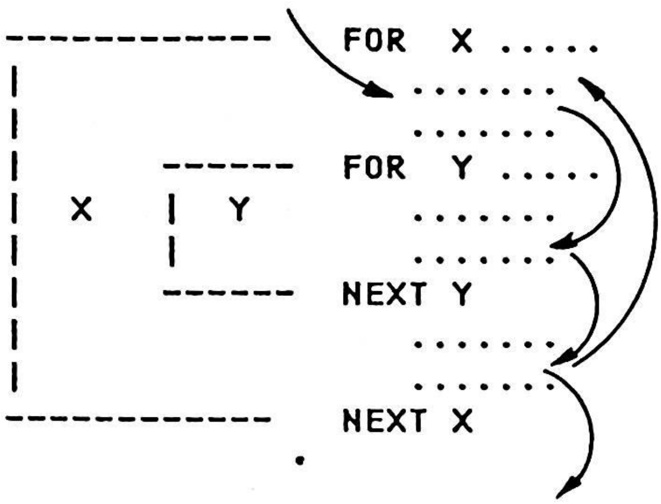
בתכנית שלפנינו מבצעת הלולאה הפנימית M הדפסה של מכפלת שני ערכים ואינה מקדמת את השורה (מכיון שרשמנו ; בסיום הוראת PRINT שבשורה 3). בצורה זו יירשמו 10 מכפלות בשורה אחת.

ההוראה PRINT שבשורה 50, לאחר סיום הלולאה, תגרום לקידום השורה (מעבר לשורה הבאה), ובפעם הבאה תבוצע הדפסה בשורה חדשה. ההוראה בשורה 6 תחזיר אותנו לתחילת לולאה, K אם המונה K לא עבר את התחום העליון שלו. כאשר נחזור לתחילת לולאה K, ערך המונה K שווה ל-2. בשורה 20 נגיע ללולאה הפנימית M, אך עתה יהיה ערך המונה M שווה ל-1. הלולאה הפנימית תבוצע עתה 10 פעמים ותדפיס בשורה אחת את המכפלות 2X1, 2X2, 2X3 וכו' עד 2X10. כאשר תסתיים הלולאה, נעבור לשורה 50, אשר תבצע דילוג לשורה חדשה בלבד. בשורה 60 תסתיים לולאה K פעם נוספת, והמונה K יקודם למספר 3. נחזור להתחלה וכן הלאה, עד אשר תסתיים לולאה K בפעם העשירית, והתכנית תמשיך להוראה END, שהיא גם הוראה STOP לסיום התכנית.

מה קרה? בכל פעם, שבוצע קידום אחד בלולאה החיצונית K, ביצעה הלולאה הפנימית מחזור שלם של לולאות, כאשר התקדמה מהערך התחתון ועד הערך העליון. תהליך זה גרם לכך שלולאה M שלטה על הדפסת המכפלות בשורה ולולאה K שלטה על קידום השורות ועל המונה בעבור המכפלה. אפשר להשוות זאת לתנועת המחוגים בשעון. בעת שמחוג השעות נע 1/12 של הסיבוב (מעבר משעה אחת לשנייה), מבצע מחוג הדקות סיבוב שלם.

<u>14.11.1 כללים נוספים לכתיבת לולאות מקוננות</u>

כאשר מתכנתים לולאות מקוננות, אפשר לבצע דילוגים בתוך תחום הלולאה בלבד. אסור לעבור מתחומי לולאה אחת לתחומי לולאה אחרת. הסיבה לכך היא, שדילוג מעבר להוראות FOR ו-NEXT גורם לשיבוש בפיקוח האינדקס. נתאר באופן סכימטי את <u>האיסורים:</u>



כל החיצים
מציינים פעולות
<u>אסורות</u>

<u>14.12 מערכים</u>

<u>מערך</u> (ARRAY) הינו דרך פשוטה לרשום אוסף נתונים בצורה, שיישמר הקשר בין נתון לנתון. טבלה, למשל, הינה דוגמה לכך. המערך יכול להיות חד-ממדי בצורת <u>וקטור</u>, כלומר שורה של מספרים, כמו למשל:

4, 1, 14, 8, 9

המערך יכול להיות דו-ממדי בצורת <u>טבלה</u> או <u>מטריצה</u>, ואז הוא מורכב משורות ומעמודות, כמו למשל:

5	6	20	45
14	24	3	80
84	30	19	37

השימוש במערכים נפוץ במיוחד בבעיות מדעיות. בעבודות אלו נמצא גם מערכים רב-ממדיים. השימוש במערכים אינו מוגבל למספרים בלבד, ובהמשך נכיר גם מערכים של ערכים אלפנומריים.

<u>14.12.1 מערך חד-ממדי - וקטור</u>

המערך הראשוני הוא <u>מערך חד-ממדי</u>, או <u>הוקטור.</u> נתון, למשל, אוסף המספרים הבא:

15 17 93 21 30

בכתיב המתמטי נסמן את הוקטור באות A, למשל, ולאיברים שלו נפנה על ידי הוספת אינדקס לשם הוקטור. כך למשל, יסמן A1 את המספר 15 בדוגמה, A3 - את 93 וכך הלאה. הכרזת הוקטור בבייסיק מכוונת להגדרת שם הוקטור ומספר האיברים שלו. למשל, הכרזת וקטור M של ערכים מספריים, שבו 5 איברים:

```
10   DIM   M(5)
```

הכרזה הבאה:

```
10  DIM  R$(4)
```

תציין וקטור R$ של ערכים אלפנומריים ובו 4 איברים. אם לא צויין אחרת, אורך כל נתון הוא 14 תווים (מספר זה עשוי להיות שונה במערכות שונות). אפשר לציין מערך עם איברים באורך 8 תווים כל אחד:

```
15   DIM B$8(4)
```

איברי המערך החד-ממדי, או הוקטור, יירשמו בהוראות בייסיק השונות בציון שם המערך, ובסוגריים יירשם המספר הסידורי של האיבר. כך למשל, M(2) יציין את האיבר השני במערך, וכדי להכניס את הערך 7 לאיבר החמישי של המערך M נרשום:

```
30   LET M(5) = 7
```

ההוראה

```
20   R$(3)="ABC53X"
```

תגרום להכנסת מחרוזת התווים שבין הגרשיים לתוך האיבר השלישי של המערך R$.

שים לב שהוראה DIM צריכה להופיע <u>לפני</u> שפונים בפעם הראשונה למערך המוכרז בה. מקובל לרשום את <u>כל</u> הוראות DIM בתחילת התכנית. אפשר להכריז על יותר ממערך אחד בהוראה, לדוגמה:

```
30   DIM A(20) ,C(150) , R(80)
```

A - מערך וקטור בן 20 איברים
C - מערך וקטור בן 150 איברים
R - מערך וקטור בן 80 איברים

לשם הדגמה נפתור בעיה פשוטה:

נתון מערך בן 10 איברים. יש להכניס בו את המספרים 51 עד 60 בהתאמה ולהציגם.

```
10   DIM K(10)              REM ARRAY DEFINITION
20   FOR L = 1 TO 10
30   K(L) = 50 + L          REM LOAD THE ARRAY
40   PRINT "ITEM ";L;" IS:"; K(L)
50   NEXT L
60   END
```

10 הגדרת מערך K בן 10 איברים.

20 הגדרת לולאה עם מונה L אשר תתבצע 10 פעמים.

30 הכנסת ערכים לאיברי המערך: הערך הינו הסכום של

מספר קבוע 50 וערך משתנה של המונה L. כלומר באיבר K(1) יוכנס הערך 50+1, באיבר K(2) - הערך 50+2, וכן הלאה עד 50+10, שהוא הערך האחרון.

40 הצגת האיברים של המערך. כל איבר בשורה נפרדת.

צורת ההדפסה :
```
ITEM  1 IS:51
ITEM  2 IS:52
ITEM 10 IS:60
```

50 סיום הלולאה.

60 סיום התכנית.

כבר ראינו, שאפשר לקדם את המונה של בקרת הלולאה בדילוגים שונים מ-1. כך למשל, אם רוצים לגרום, לכך שכל האיברים הזוגיים בטבלה בת 50 איברים יוכפלו ב-3 (בהנחה שיש שם נתונים מפעולה קודמת), אפשר לכתוב:

```
10    DIM R(50)
20    FOR I = 2 TO 50 STEP 2
30    R(I) = R(I)*3
40    NEXT I
50    END
```

10 הכרזת טבלה בת חמישים איברים.

20 קביעת הלולאה. מונה הלולאה I יתקדם ב-2, החל מהמספר 2.

30 מכפלת ערך האיבר בטבלה ב-3 ואחסנה חוזרת באותו איבר.

40 סיום הלולאה.

50 סיום התכנית.

<u>טעינת נתונים</u> למערך יכולה להיות ע"י שימוש בהוראת לולאה והוראת READ. כל פעולה READ תגרום לטעינת נתון באיבר, שמספרו הסידורי הוא I.

```
10    DIM K(3)
20    FOR I = 1 TO 3
30    READ K(I)
40    NEXT I
50    DATA 14,3,9
60    END
```

אפשר **להזין נתונים** למערך מחוץ לתכנית, על ידי המפעיל. כל פעולת INPUT תגרום לטעינת הנתון לאיבר העוקב בטבלה:

```
10   DIM K(3)
15   FOR E = 1 TO 3
16   INPUT K(E)
17   NEXT E
```

תרגיל

נתונה טבלה באורך 50 איברים. יש למצוא את הערך הגבוה ביותר של אחד מאיברי הטבלה ולהעבירו למשתנה B, ואת מספרו היחסי בטבלה יש לרשום במשתנה M.

```
 10   DIM A(50)
 20   B = A(1)
 30   M = 1
 40   FOR I = 2 TO 50
 50   IF B > A(I) GOTO 80
 60   B = A(I)
 70   M = I
 80   NEXT I
 90   PRINT "THE LARGEST NUMBER:";B,"LOCATION:";M
100   END
```

10 הגדרת מערך בן 50 איברים. אנו מניחים, שבמערך יש ערכים, שהוטענו בו קודם.

30,20 כתנאי התחלה מניחים שהאיבר הראשון הוא הגבוה ביותר, ולכן ערכו הוכנס ל-B ומספרו הסידורי הוכנס ל-M. (אפשר להתחיל, למשל, עם B=-99999, שהוא המספר הקטן ביותר שניתן להציג במחשב, ואז ידרש סיבוב בדיקה אחד נוסף בעבור האיבר הראשון, אשר בוודאות יהיה גדול ממנו).

40 בניית הלולאה. המונה I ישתנה מ-2 עד 50 בדילוגים של 1. התחלנו ב-2 כי אין צורך להשוות את הערך הראשון שבטבלה לעצמו.

50 השוואת ערך האיבר ששמרנו לערך האיבר התורן A (I) שבטבלה. אם הערך של B גדול או שווה לאיבר בטבלה, עוברים מיד להוראה האחרונה בלולאה בשורה 80, ומכאן לתחילת הלולאה, אם המונה I טרם הגיע לערך הגבוה שלו (50).

70,60 אם התנאי בהוראה IF לא מתמלא, כלומר האיבר בטבלה

גדול בערכו מהאיבר ששמרנו ב-B, יש לטעון ל-B את הערך החדש, ואת המספר הסידורי של האיבר בטבלה יש לטעון ל-M.

80 הוראת הסיום של הלולאה המחזירה לראש הלולאה, אם המונה טרם הגיע לערכו הגבוה.

90 הדפסה של המספר הגדול ומיקומו הסידורי במערך.

100 סיום התכנית.

14.12.2 מערך דו-ממדי - טבלה

המערך הדו-ממדי קרוי גם בשם טבלה או מטריצה. ישנן מטריצות או טבלאות בעלות ממדים רבים יותר, אך לא נעסוק בהן כאן. הטבלה מורכבת ממספרים או ערכים אלפנומריים הערוכים בשורות ובעמודות ויש לה שם מסוג משתנה: שם משתנה B, למשל, יציין טבלה, המכילה ערכים נומריים, ו-P$ יציין טבלה של ערכים אלפנומריים.

על גודל טבלה מכריזים באמצעות ציון מספר השורות ומספר העמודות שלה, כמו למשל, 10 DIM B (5,12). בטבלה B מוגדרות 5 שורות ו-12 עמודות, או סה"כ 60 איברים. נפנה לאיבר בטבלה על-ידי ציון מספר השורה ומספר העמודה שלו, למשל,

```
40 B(8,2) = C + 3
```

הסכום C+3 יוטען לאיבר בטבלה B, אשר נמצא בשורה 8 ובעמודה 2.

בהוראה אחת אפשר להגדיר יותר ממערך אחד:

```
30 DIM A(5,8), C(12), K(5,6,3)
```

A - מערך דו-מדדי (טבלה) של 5 שורות ו-8 עמודות,
C - מערך חד-ממדי (וקטור) בן 12 איברים,
K - מערך תלת-ממדי ("תיבה") שבו 5 שורות ו-6 עמודות, אשר חוזרות 3 פעמים. קובייה, אשר הצלע שלה 4 איברים תסומן B(4,4,4).

מצייני האיברים של הטבלה יהיו על פי רב משתנים או ביטויים ולא ערכים קבועים, ובכך אפשר להיות גמישים יותר בתכנות הבעייה. נוכל לציין, למשל:

```
10   LET S = C(M*3) + B(I)
50   LET M$ = R$(7,I+2)
80   IF S(5,A*4) = B(K) GOTO 170
```

העברה ממערך חד-ממדי למערך דו-ממדי תיעשה בצורה הבאה: האיברים מהמערך החד-ממדי יועברו בזה אחר זה, ובטבלה תתמלאנה **השורות** בזו אחר זו, בראש ובראשונה תתמלאנה העמודות בשורה הראשונה, ואח"כ העמודות של השנייה וכן הלאה.

נתון מערך M(6), כלומר וקטור בן 6 איברים, ומערך B(2,3) שהוא מטריצה של 2 שורות ו-3 עמודות. כאשר נעביר את איברי M לאיברי B, נקבל:

```
   מערך M           M(1) = 1     B(1,1) = 1          מערך B
   ------           M(2) = 2     B(1,2) = 2          ------
1  2  3  4  5  6    M(3) = 3     B(1,3) = 3           1 2 3
        ------>     M(4) = 4     B(2,1) = 4   ---->   4 5 6
                    M(5) = 5     B(2,2) = 5
                    M(6) = 6     B(2,3) = 6
```

הזנת נתונים לטבלה יכולה להיעשות מתוך רשימת נתונים, שהוגדרו בהוראת DATA באמצעות הוראת READ, או מן המפעיל של המחשב באמצעות הוראת INPUT.

```
10   DIM P(2,3)
20   FOR I=1 TO 2
30   FOR K=1 TO 3
40   READ P(I,K)
50   NEXT K,I
60   DATA 12,3,8,4,7,16,9,15
70   END
```

הפלט של התכנית היא טבלה P, אשר תראה כך:

```
12  3   8
 4  7  16
```

10 הכרזת מערך, טבלה דו-ממדית.

20 לולאת קידום השורות.

30 קריאה לאיברי הטבלה מתוך רשימת הנתונים שהוכרזה בהוראה DATA, בשורה 60. המספרים 8,3,12 יוזנו בעת הביצוע של הלולאה K במחזור הראשון אל האיברים P(1,1) P(1,2) P(1,3) בהתאמה. בסיבוב השני יוזנו המספרים 16,7,4 אל האיברים P(2,1),P(2,2),P(2,3).

50 מכיון שיש סיום חופף לשתי הלולאות, השתמשנו בהוראה NEXT אחת. שים לב, שסדר רישום האינדקסים תואם את

סדר ביצוע הלולאות. קודם K ואח״כ הלולאה החיצונית I.

60 רשימת הנתונים. יש יותר איברים מן הדרוש לתכנית אך אין זה חשוב.

70 סיום התכנית.

הזנת נתונים לטבלה ע״י המפעיל בהוראה INPUT תיעשה בצורה דומה:

```
10   DIM P(2,3)
20   FOR I = 1 TO 2
30   FOR K = 1 TO 3
40   INPUT "ONE NUMBER PLEASE:";P(I,K)
50   NEXT K,I
60   END
```

תכנית זו זהה לתכנית הקודמת. ההוראה INPUT בשורה 40 באה במקום ההוראה READ (שים לב, שרצוי לתת הנחייה, מה רוצים לקבל כקלט). ההוראה DATA להגברת נתונים הושמטה, כי הקלט הוא ממקור חיצוני.

14.12.3 הוראת עיבוד לטבלה - MAT

הוראה מיוחדת הקשורה לטבלאות היא MAT. הוראה זו מאפשרת התייחסות לטבלאות שלימות בלי לציין במפורש את האיברים השונים שלהן. להוראה שני שימושים:

- טיפול בנתונים של טבלה, שהוגדרה קודם לכן בהוראה DIM

הגדרת טבלה	10	DIM K(2,3)	--->	000 000
טעינת ערך אחד	20	LET K(2,1)=7	--->	000 700
טעינת ערכים לטבלה שהוגדרה קודם	30	MAT K=5	--->	555 555

- הגדרת טבלה וטיפול בנתונים שבה בהוראה אחת.

```
50  MAT A(2,4)=9
```

הגדרת טבלה בת 2 שורות ו-4 עמודות וטעינת המספר 9 <u>לכל</u> האיברים.

בהוראת MAT אחת אפשר לפנות לטבלאות אחדות.

טעינת נתונים למערך באמצעות הוראה MAT יכולה להיעשות מתוך רשימת נתונים, המוגדרים בתכנית באמצעות DATA או מן המפעיל באמצעות INPUT.

14.12.4 הזנת נתונים בהוראה MAT READ

הזנת נתונים בהוראה זו אינה שונה מהזנת נתונים בהוראת READ רגילה. ההבדל הוא, בכך שכאן תגרום ההוראה לטעינת כל הטבלה ולא לטעינת משתנה אחד בלבד. הטבלה נטענת לפי שורות בזו אחר זו. נראה, כיצד נעשה הדבר בתכנות:

```
10   DIM K(2,4)
20   DATA 2,17,3,4,7,9,8,13   REM 8 NUMBERS REQUIRED
30   MAT READ K
```

הטבלה שתבנה:

```
2  17  3  4
7   9  8  13
```

אם במאגר DATA יהיו פחות נתונים מהנדרש, הטבלה לא תתמלא ותנתן הודעת שגיאה. אם יהיו יותר נתונים מהנדרש, ההוראה MAT לא תטפל בהם.

14.12.5 הזנת נתונים בהוראה MAT INPUT

נשתמש בדוגמה, כדי להסביר את הדרך, שבה נטענת הטבלה. הטעינה היא גם כאן בשורות. כל הוראת קלט מבקשת הזנת נתונים, והמפעיל מזין אותם בעבור שורה אחת מידי פעם. נשתמש הפעם בטבלה של ערכים אלפנומריים.

```
20   DIM R$(2,3)
30   MAT INPUT R$
```

?	המחשב יבקש נתונים:
A,B,4	המפעיל יזין (המספר 4 הוא "אלפנומרי")
??	בעבור השורה השנייה המחשב יבקש
C,D,E	

בזה יסתיים התהליך. עבור כל שורה נוספת, אם נדרש, יבקש המחשב נתונים באמצעות שני סימני שאלה (??).

14.12.6 פעולות במערכים

אפשר לבצע במערכים פעולות שונות.

נגדיר 3 מערכים

```
10   DIM X(5),Y(5),Z(5)
```

נוכל לחבר את שני המערכים `50 MAT X=Y+Z`

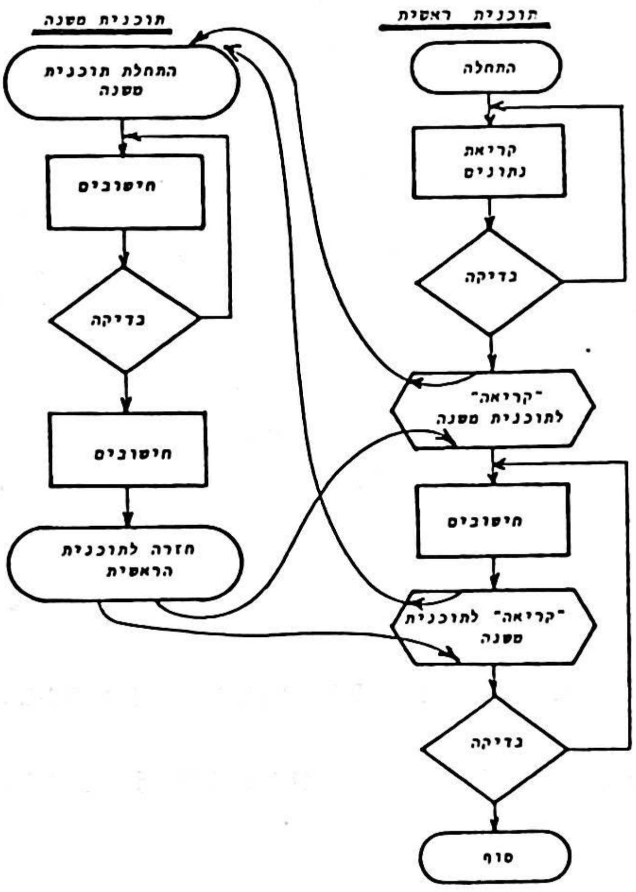
או לחסר אותם `80 MAT X=Y-Z`

מספר האיברים שווה בכל המערכים, והפעולה תבוצע בין האיברים השונים בהתאמה. אפשר לכפול מערך במערך, למיין, להפוך ולמעשה לבצע את כל הפעולות הידועות מן המתמטיקה. הנושא חורג מתחום הדיון שלנו. קורא, המעוניין בכך, ימצא הסבר מפורט בספרות המקצועית, המתייחסת למחשב, שבו הוא עובד.

14.13 תכניות משנה

תכנית עיבוד מורכבת מקטעים שונים, המבצעים משימות שונות בתהליך העיבוד. קטעים מסויימים בתכנית עשויים לחזור על עצמם, כמו למשל, הדפסת כותרת בראש כל דף, חישוב שורש ריבועי, חישוב ריבית, חישוב מס-הכנסה בתלוש שכר, מציאת הערך הגדול מתוך ערכים נתונים ועוד. גם אם קטע מסוים איננו חוזר על עצמו באותה תכנית, נמצא, שהוא עשוי לחזור ולהופיע בתכניות אחרות. כדאי, אם כן, לשמור על אותם קטעי תכניות ולהשתמש בהם שוב ושוב ללא צורך בכתיבה מחדש.קטעים אלו הם <u>תכניות משנה</u> (SUBPROGRAMS), אשר נשמרות <u>בספריית תכניות</u> (PROGRAM LIBRARY) והתכנית הראשית "קוראת" להן. תפקידו של המתכנת לדאוג לשילוב הלוגי של תכניות המשנה בתכניתו, <u>והמהדיר</u> (COMPILER) או <u>המפענח</u> (INTERPRETER) מטפל בשילוב הפיסי שלהן בתכנית. ספריית תכניות המשנה עומדת לרשותו של כל מתכנת, ועל כן הוא פטור מלכתוב את אותן התכניות מחדש, אם הן מתאימות לצרכיו.

נתבונן בנוהל השילוב של תכניות המשנה ובשיטה בה מבוצע העיבוד בעזרת התהליך הסכימטי הבא:



בעת ההפעלה של התכנית הראשית, מבוצעות ההוראות השונות בזו אחר זו, עד אשר מגיעים להוראת "קריאה" או סיעוף לתכנית משנה. בנקודה זו מבוצע מעבר להוראה הראשונה בתכנית המשנה, וממנה ממשיכים בביצוע ההוראות בזו אחר זו, עד אשר מסיימים. עתה חוזרים אל התכנית הראשית, אל ההוראה העוקבת את הוראת הקריאה, לתכנית המשנה. מכאן ממשיך תהליך העיבוד בתכנית הראשית עד לסיומה, או עד אשר פוגשים שוב בהוראת קריאה לתכנית משנה זו, או לתכנית משנה אחרת.

קיימים סוגים אחדים של תכניות משנה, <u>פונקציות ושגרות,</u> וגם בשתי קבוצות אלו יש חלוקה נוספת. עתה נראה, כיצד משתמשים בפונקציות ושגרות, וכיצד כותבים אותן, כאשר דרוש.

<u>14.13.1 פונקציות בנויות או פונקציות ספרייה</u>

כל העוסק בתכנות מדעי, נתקל לעתים קרובות במספר פונקציות כמו הפונקציות הטריגונומטריות, לוגריתמים, שורש ריבועי, מספר אקראי ועוד. אולם אין הפונקציות האלו משרתות תכנות מדעי בלבד, הן שימושיות גם בנושאים אחרים. מלבד זאת יכול המתכנת להגדיר פונקציות נוספות בהתאם לכללי הכתיבה של הפונקציות.

כדי להקל על המתכנת, מצורפת לבייסיק ספריית תכניות, שכוללת מספר רב של פונקציות, אשר אפשר לשלבן בצורה פשוטה בתוך התכנית, אלו הן <u>פונקציות בנויות או פונקציות ספרייה</u> (INTRINSIC FUNCTIONS). הפנייה של התכניות אל פונקצייה בנויה נעשית ע"י ציון שמה והערכים (פרמטרים) הנדרשים לצורך החישוב.

נסקור עתה מספר פונקציות נפוצות וצורת כתיבתן (שים לב, שבמחשבים שונים תתכן צורת כתיבה שונה):

מציאת ערך מוחלט של משתנה X	ABS(X)
שעה, במבנה HH:MM:SS	CLK
חישוב סינוס של X (ברדיאנים)	SIN(X)
מעריך טבעי של X	EXP(X)
מספר התווים במחרוזת X$	LEN(X$)
המספר המקסימלי בקבוצה X,Y...	MAX(X,Y...)
שורש ריבועי של X	SQR(X)
קבלת מספר אקראי בתחום 0 עד 1	RND
סכום האיברים במערך X	SUM(X)

כשם שישנן שתי קבוצות של נתונים: נתונים מספריים ונתונים אלפנומריים, קיימות גם שתי קבוצות של פונקציות: פונקצייה מספרית ופונקצייה אלפנומרית, בהתאם לערך שהיא מחזירה.

הפונקצייה מקבלת ערך בודד: קבוע, משתנה או ביטוי (אשר מחושב אוטומטית). היא "מחזירה" כפלט ערך יחיד, אשר אפשר להשתמש בו

בתוך ביטוי, במקום שהפונקצייה רשומה, או להעביר אותו למשתנה. בצורה זו הפונקצייה הינה הרחבה של "הביטוי":

```
 60   V = SIN(1)
 70   LET V = SIN(Y)
 80   LET Z = 1 + SQR(X**3)
 90   W = 1 - SQR(COS(A))
100   T$ = CLK
```

אפשר לכתוב פונקצייה, שמקבלת ערך של ביטוי, אשר מחושב, לפני שהפונקצייה מתחילה לפעול (שורה 80) והיא עצמה חלק מביטוי. אפשר לכלול פונקצייה בתוך פונקצייה (שורה 90). הפונקצייה CLK מחזירה ערך אלפנומרי, ולכן מציבים אותו במשתנה אלפנומרי T$ (שורה 100).

14.13.2 פונקציות מתוכנתות

כל מתכנת יכול להוסיף לתכניתו פונקציות לשימוש בתכנית זו בלבד או להוסיף פונקציות לספריית התכניות. עליו לנהוג לפי כללי הכתיבה של פונקציות בעבור ערכים מספריים או אלפנומריים. כללי השימוש בפונקציות מתוכנתות זהים לכללי השימוש בפונקציות בנויות.

1. שם פונקצייה יתחיל תמיד באותיות FN, ולאחריהן תהיה האות בעבור פונקצייה לטיפול בערכים מספריים. בפונקצייה לטיפול בערכים אלפנומריים יש להוסיף את הסימן $, כמקובל במשתנים אלפנומריים.

   FNA - שם של פונקצייה לטיפול בערכים מספריים.
   FNK$ - שם של פונקצייה לטיפול בערכים אלפנומריים.

2. מגדירים פונקצייה בהוראה DEF, כשלאחריה נרשמים בסוגריים פרמטר אחד או יותר. הפונקצייה מחזירה תמיד ערך אחד סופי. את ההגדרה רושמים על פי רב בתחילת התכנית, לפני שמשתמשים בפונקצייה.

```
50   DEF FNA(X)
```

3. פונים אל הפונקצייה ע"י הכרזת שמה, ואחריה, בסוגריים, תופיע רשימת הפרמטרים. הפונקצייה מחזירה ערך יחיד, שסוגו ושמו כשם הפונקצייה (ערך מספרי או אלפנומרי). הפרמטרים, שנמסרים לפונקצייה, צריכים להתאים בסוגם לדרישות הפונקצייה: מספרים או ערכים אלפנומריים, שמות של משתנים או מערכים ועוד. שמות המשתנים והמערכים אינם זהים בדרך כלל לאלו, שמוגדרים כפונקצייה, ולכן <u>חשובים</u> הסוג, מספר הפרמטרים והסדר שלהם.

4. אפשר להוסיף לאחר ההגדרה סימן שוויון (=) ומימין לו

ביטוי אריתמטי אשר מייצג את תהליך העיבוד כנדרש, ואפשר לכלול בביטוי פונקצייה אחרת.

```
60    DEF FNB(R) = R + K*3
70    DEF FNA(D) = SQR(S*4-5)
```

5. אפשר לכתוב פונקצייה, אשר בנויה ממספר ביטויים. במקרה זה יש להשתמש בהוראה RETURN (ולא STOP) עם שם משתנה, כדי לחזור לתכנית הראשית. בסוף ההוראות יש לרשום FNEND.

<u>תרגיל</u>

נשתמש בכללים אלו לכתיבת פונקצייה לחישוב ממוצע הציונים של תלמידי כיתה י"ב. המורה יזין למערכת בעבור כל תלמיד את מספר התלמיד והציונים של 4 מקצועות הלימוד הרצויים. התכנית תחשב ותציג את ממוצע הציונים של כל תלמיד. מכיון שהתכנית תיכתב בצורה כללית, לא נציין מראש את מספר התלמידים. המורה יקיש אפס (0) במקום מספר התלמיד, כדי לציין את סיום הרשימה. התכנית תספור את מספר התלמידים כדי לחשב את הממוצע.

<u>התכנית הראשית</u> תטפל בקליטת הנתונים מהמסוף ובהדפסת תוצאות החישוב. הפונקצייה תבצע חישוב של ממוצע 4 הציונים ותמסור את התוצאה לתכנית הראשית להצגה. הפונקצייה הזו פשוטה, ואפשר לכתוב אותה בשורה אחת.

```
 10   DEF FNA (A,B,C,D) = (A+B+C+D)/4
 20   K=0
 30   PRINT "STUDENT ID AND AVR RATING"
 40   INPUT "TYPE ID AND 4 NUMBERS";N,W,X,Y,Z
 50   IF N=0 GOTO TO 90            REM CHK END OF DATA
 60   PRINT N;"-";W;X;Y;Z;"-";FNA(W,X,Y,Z)
 70   K = K+1
 80   GOTO 40
 90   PRINT "NUMBER OF STUDENTS-";K
100   END
```

ההדפסה של התוצאה (שורה 60) תהיה: 5783-80 60 85 75-75.

<u>הסבר</u>

10 שם הפונקצייה הוא FNA, והיא מקבלת 4 פרמטרים, שהם 4 מספרים. תוצאת החישוב של הפונקצייה נמסרת במשתנה, ששמו כשם הפונקצייה.

20 איפוס מונה לספירת מספר התלמידים.

30 הדפסת כותרת לדו"ח.

40 הדפסת בקשה להזנת נתונים: מספר התלמיד ו-4 ציונים.

50 בדיקה אם שדה מספר התלמיד שווה לאפס. אם כן - דילוג לפעולות הסיום של התכנית.

60 הצגת מספר התלמיד ורשימת הציונים והממוצע (פונקצייה). בין כל קבוצת נתונים יהיה קו מפריד.

70 ספירת התלמידים, ע"י הוספת 1 למונה.

80 חזרה להוראת הקריאה של נתונים חדשים.

90 בסיום התכנית, הצגת מספר התלמידים.

100 סיום התכנית. אם לא נרשמה הפקודה STOP, תשמש הפקודה END למטרה זו.

פונקצייה עשויה להיות מורכבת יותר, ואז היא תירשם כקבוצת הוראות ולא כהוראה אחת. ההוראה RETURN מכוונת להחזיר את תוצאת החישוב של הפונקצייה אל התכנית הראשית, אשר קראה לה. הפונקצייה הקודמת לחישוב הממוצע יכולה להיכתב גם כך:

```
11    DEF FNA (A,B,C,D)
12    R = (A+B+C+D)/4
13    RETURN R
14    FNEND
```

או למשל

```
17    DEF FNA(A,B,C,D)
18    RETURN (A+B+C+D)/4
19    FNEND
```

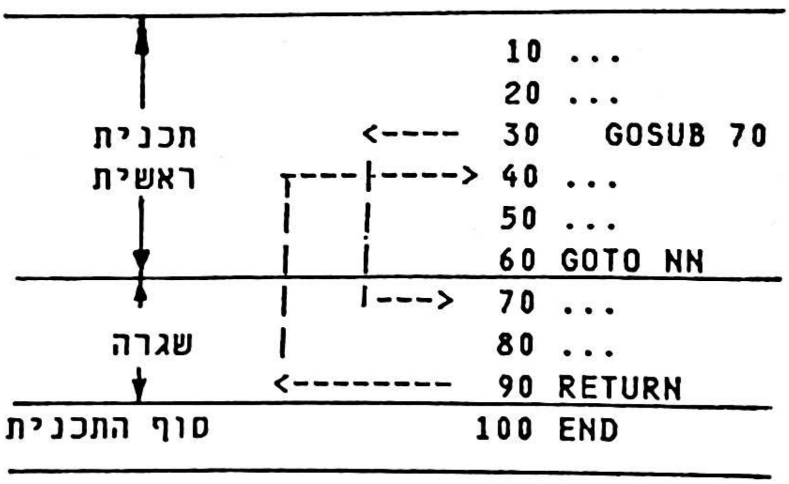
קבוצות הוראות אלו תירשמנה במקום שורה 10 בדוגמה הקודמת. השימוש בפונקצייה יהיה כמו במקרה הקודם.

<u>14.13.3 שגרה</u>

<u>שגרה</u> (SUBROUTINE) היא תכנית, אשר משולבת בתכנית הראשית, ומטרתה לבצע סדרת הוראות מוגדרות. משתמשים בשיטת תכנות זו, כאשר יש לפנות ממקומות שונים בתכנית הראשית אל סדרת הוראות זו. אפשר לפנות אל שגרה גם מתוך שגרה אחרת. אפשר לראות את השגרה כהרחבה של מושג הפונקצייה מבחינת גמישות השימוש והאפשרויות. בניגוד לפונקצייה, משתמשת השגרה במשתנים של התכנית, שבה היא נמצאת.

הוראות הדילוג מהתכנית הראשית הן אל ראשית השגרה. הוראות הדילוג מתוך השגרה אינן יכולות לצאת ממנה אל התכנית הראשית,

כדי שלא לאבד את הרצף של ביצוע התכנית, אך אפשר לדלג אל הוראת הסיום של התכנית (STOP). המעבר מרצף ההוראות בתכנית לביצוע שגרה הוא על ידי ההוראה GOSUB NN, כאשר המספר NN מבטא את מספר השורה, שבה מתחילה השגרה. ההוראה האחרונה בשגרה תהיה RETURN, אשר תחזיר את הפיקוח להוראה, העוקבת את ההוראה GOSUB. נדגים זאת בתרשים:



NN - מספר שורה בתכנית הראשית

20,10 הוראות התכנית הראשית.

30 דילוג אל שגרה המתחילה בשורה 70.

50,40 הוראות התכנית הראשית.

60 הוראת התכנית הראשית. זוהי הוראת דילוג או הוראת STOP.

70 ההוראה הראשונה של השגרה.

80 הוראת השגרה.

90 ההוראה האחרונה בשגרה (אינה חייבת להיות האחרונה מבחינת הסדר אלא מבחינת הביצוע). היא מחזירה את הפיקוח <u>להוראה העוקבת</u> להוראת GOSUB אשר ביצעה את הדילוג לשגרה.

100 סוף הוראות התכנית.

<u>תרגיל</u>

נדגים את השימוש בשגרה בתכנית, אשר מוצאת את המכנה המשותף הגבוה ביותר שבין 3 ערכים, אשר מוזנים מן המסוף. התוצאה נמסרת במשתנה G.

```
<--------  50   GOSUB 200
|          60   X = G
|          70   Y = C
|   <---   80   GOSUB 200
|   |      85   PRINT "A","B","C","CD"  REM PRINT IN STD COLUMNS
|   |      90   PRINT A, B, C, G
|   |     100   STOP
-------->  200   Q = INT (X/Y)           REM Q IS INTEGER OF X/Y
```

```
 10   PRINT "ENTER THREE INTEGERS"
 20   INPUT A, B, C
 30   X = A
 40   Y = B
 50   GOSUB 200
 60   X = G
 70   Y = C
 80   GOSUB 200
 85   PRINT "A","B","C","CD"            REM PRINT IN STD COLUMNS
 90   PRINT A, B, C, G
100   STOP
200   Q = INT (X/Y)                     REM Q IS INTEGER OF X/Y
210   R = X - Q*Y
220   IF R = 0 GOTO 260
230   X = Y
240   Y = R
250   GOTO 200
260   G = Y
270   RETURN
300   END
```

הסבר התכנית:

10 הנחייה להזנה של 3 נתונים.

20 הוראת הקלט ל-3 מספרים C,B,A (אפשר לשלב את הוראות 10 ו-20, איך?).

40,30 העברה של ערכי B,A ל-Y,X בהתאמה, כדי שבמהלך החישובים לא "נקלקל" את הערכים המקוריים.

50 עתה עוברים לשגרה המתחילה בשורה 200. היא "מקבלת" שני ערכים X ו-Y.

60 חזרה להוראה זו לאחר הביצוע הראשון של השגרה. במשתנה G ישנו המכנה המשותף הגדול ביותר שבין X ו-Y כלומר בין A ו-B. נעביר אותו למשתנה X כדי שנוכל להמשיך לחשב.

70 את משתנה C נעביר למשתנה Y.

80 נעבור פעם נוספת לשגרה לאחר הכנת המשתנים X ו-Y.

85 הצגת כותרת לתוצאות החישוב.

90 הצגת הערכים המקוריים C,B,A והתוצאה G.

100 הוראת עצירה, סיום ביצוע התכנית ומעבר למערכת ההפעלה.

ואת המשתנה הנוסף C. בתוך השגרה יש סיעוף אל הוראת ההתחלה שלה באמצעות הוראת GOTO (שורה 250).
בסיום השגרה תמסר התוצאה במשתנה G.

300 סוף התכנית.

הביצוע של התכנית

כאשר נפעיל את התכנית, היא תדפים:

```
ENTER THREE NUMBERS
?
60, 90, 120
```

נזין את הנתונים: 60, 90, 120

הפלט: כותרת

```
A    B    C     CD
60   90   120   30
```

מספרים

הרווחים שבין עמודות ההדפסה יהיו בהתאם למחשב, שבו תופעל התכנית. במחשב יבמ, למשל, העמודות הן: 1, 19, 37, 55 בהתאמה.

14.14 הדפסה לפי תבנית מוגדרת - PRINT USING

הוראת ההדפסה PRINT, שהשתמשנו בה עד עתה, גרמה להדפסת מספרים או ערכים אלפנומריים בשדות מוגדרים קבועים מראש או בשדות צמודים (כשהשתמשנו ב-; להפרדה), והם הודפסו במלואם מתוך המשתנה. לדוגמה, אם משתנה, שהוא תוצאת חישוב של מחיר פריט, הכיל 10 ספרות ומהן 8 הן השבר העשרוני, הודפס המספר בשלימותו, גם אם אין כל משמעות לרוב הספרות העשרוניות. נרצה, למשל, להדפים שתיים מהן בלבד, בתבנית 78.30, למשל, ולא 78.30634762. ייתכן גם, שנרצה לעגל את המספר, אך זו בעייה אחרת.

בקרת התבנית של ההדפסה נעשית באמצעות ההוראה הבאה:

```
PRINT USING NN, V1, V2...
```

NN מבטא מספר שורה, שבה מוגדרת <u>תבנית ההדפסה</u> הרצויה (IMAGE או FORM), V1, V2 וכו', מציינים את המשתנים שיודפסו, כמו בהוראת PRINT רגילה. נרשום את ההוראה הבאה:

```
70 PRINT USING 85, S, T
85 :ITEM ### COSTS ###.## SHEKEL
```

ההדפסה תבוצע לפי שורת התבנית מספר 85. המשתנים שיודפסו הם S, T (לפי הסדר).

שורת התבנית מתחילה בסימן הנקודתיים : כדי להבדילה משורות הוראה אחרות. השדה המסומן ב-# מיועד להכנסת ערך של משתנה. אורך השדה הראשון 3 תווים, והשדה השני הוא בן 6 תווים, שמהם 3 תווים מיועדים לחלק השלם של המספר, אחד לנקודה עשרונית ו-2 תווים מיועדים לחלק העשרוני של המספר.

אם T=425.705, S=649 נקבל את השורה המודפסת הבאה:

```
ITEM 649 COSTS 425.70 SHEKEL
```

הערך של המספר יוצב בצורה נכונה בהתאם למיקום של הנקודה העשרונית בתבנית. שים לב, לכך שהרווחים שבין המלים והרווחים שבין המלים לבין התבנית נשארו ללא שינוי. כך למשל, המלה SHEKEL נמצאת במרווח של תו ריק אחד מהספרה 3, מכיון שגם בשורת המבנה 85 היה רווח של תו אחד בין שדה המספר שסומן ב-# לבין המלה SHEKEL.

שים לב, שתבנית השדה כללה את סימן הנקודה העשרונית, כדי שהמספר יודפס כראוי. היא יכולה להכיל גם סימנים נוספים לעריכה שדות מספרים. סימן # לעריכת ההדפסה הוא הסימן הפשוט והחשוב ביותר. לעריכת שדות מספריים משתמשים בתבנית PIC (קיצור של PICTURE), אשר נדגים את השימוש בה בדוגמאות אחדות. סימני העריכה של תבנית PIC אינם נהוגים בכל ניבי בייסיק, או שהם שונים בניבים שונים של השפה, ולכן יש לעיין בספרות המתאימה למחשב, שבו פועלים.

נניח, שנתון המספר 123, אשר יש להדפיסו לפי תבניות PIC שונות.

התבנית	הפלט	
PIC (####)	0123	
PIC (ZZZZ)	123	
PIC (***#)	*123	
PIC (+++##)	+123	
PIC (+ZZ##)	+123	
PIC (Z/#/##)	1/23	
PIC (ZZ,#,#)	1,2,3	
PIC (ZZ,.#)	12,3.0	(נקודה עשרונית)

עתה אפשר לכתוב תכנית קצרה להדגמת השימוש בהוראה:

```
10   A = 1234.56789
20   PRINT USING 30, A
```

30 FORM PIC (ZZZ,ZZZ.##)

התוצאה שתודפס: 1,234.56

שים לב, שההוראה PIC מופיעה בשורת מבנה, המתחילה במלה FORM. אין זו הוראת ביצוע, והיא אינה "מפריעה" למהלך התכנית. רצוי לרשום את הוראות PIC ליד הוראות ההדפסה המתאימות, או לרשום את כולן יחד בראשית או בסוף התכנית.

14.14.1 פיקוח על עריכת ההדפסה בדו"ח

עריכת דו"ח מחייבת הצבת שורות ועמודות במקומות קבועים, ולא כפי שנקבע כברירת מחדל של שפת התכנות. גם שיטת תבנית # או PIC אינה מתאימה לכל הצרכים.

כשנרצה להדפיס כותרת לדו"ח במקום מסוים בראש הדף, יהיו בכותרת נתונים שונים בשמאל, באמצע ובימין. אח"כ תופיע שורת כותרת של עמודות הדו"ח, ובה שמות המתאימים לרוחב של כל עמודה ועמודה. שורות ההדפסה של הנתונים תודפסנה גם כן במבנה מוגדר, חלקן בזו אחר זו וחלקן ברווחים, ועוד. לדוגמה מבנה של כותרת דו"ח, כותרת העמודות ושתי שורות תוכן.

```
THE BASIC COMPANY INVENTORY STATUS PAGE 1 DEC.31,1983

 CAT.NO.     PART NAME    U/M   PRICE    QTY    ORDERS
 ------      ---------    ---   -----    ---    ------

  1234       BOLT         UN     5.30    735     100
  4597       CEMENT       K9
```

הוראות לבקרת ההדפסה דרושות כדי לתת למתכנת גמישות רבה יותר בעיצוב הדו"ח ובעריכת הנתונים המופקים כפלט בתהליך העיבוד. סימני הבקרה החשובים ביותר הם SPC (או X), TAB (או POS) ו-SKIP.

SPC (SPACE) - לקידום המיקום של הדפסת הנתון הבא. לדוגמה SPC(8) מורה, שיש להתקדם 8 עמודות מהמקום הנוכחי בשורה כדי להדפיס את השדה או הנתון הבא כלומר להשאיר רווחים.

TAB (TABULATOR) - מורה, שיש להדפיס את הנתון החל מעמודת הדפסה מסוימת. לדוגמה: TAB(25) מורה, שיש להדפיס את הנתון הבא החל מעמודה 25. (ספירת העמודות מתחילה תמיד בשמאל השורה).

SKIP - מורה, שיש לדלג שורות, ולמשל SKIP4 יגרום לדילוג על

שלוש שורות ולהדפסה החל מהשורה הרביעית. SKIP או SKIP1 יגרום לקידום לשורה הבאה.

את השימוש בהוראות הבקרה להדפסה נדגים ע"י הדפסת שורה מן הדו"ח שהוצג קודם. המספרים מעל השורה מציינים את עמודות ההדפסה, והמפרידים מציינים <u>כאן</u> תווי רווח ריקים (לשם הדגמה).

```
1       7               19      27    33
1576    SCREW       UN  XX7.50  XX45  XX50
```

את הנתונים להדפסה נקבל כקלט, ונדפיס אותם במבנה הנדרש:

```
10    INPUT 'NEW ITEM:',A,B$,C$,D,E,F
20    PRINT USING 30,A,B$,C$,D,E,F
30    :####--#######--##-ZZ0.##--###0-####
```

נניח שרוצים להדפיס דו"ח על שלושה שדות בלבד: מספר קטלוגי, שם הפריט והמחיר. כאשר הנתונים מוצבים בעמודות הבאות לדוגמה:

```
10      20      35
1576    SCREW   XX7.50
```

אפשר לכתוב את הוראות ההדפסה בצורה נוחה יותר מאשר עם שורת תבנית.

```
45    PRINT SPC(10);A:TAB(20);B$;TAB(35);D
```

ואפשר גם לכתוב כך בניב אחר של השפה

```
50    PRINT USING 60,A,B$,D
60    FORM X10, PIC(####),POS20,C,POS35,PIC(ZZ0.##)
```

<u>14.15 קבצים</u>

נתונים משמשים כחומר הגלם לתכניות כדי להפיק את המידע הדרוש. את הנתונים יכולה התכנית לקבל ממקורות אחדים: מתוך קבועים, אשר מוגדרים בתוכה בעת כתיבתה, כקלט מהמפעיל באמצעות לוח הקלידים או מתכנית אחרת, אשר משולבת בה. את כל האפשרויות האלו בחנו ואף תרגלנו. ראינו גם את אמצעי הפלט בתצוגה או במדפסת. לא ראינו עדיין, כיצד אפשר לפעול בכמות גדולה של נתונים, אשר נאגרת למטרת עיבוד במחשב. נתונים אלו יכולים להתקבל בתהליכי עיבוד שונים, ואפשר לצבור אותם בקובץ פלט, אשר ישמש אח"כ כקובץ קלט. במקרים מסוימים מכינים מראש קובץ ובו הנתונים הדרושים, לפני שניגשים לביצוע של תהליך עיבוד כלשהו, למשל, קובץ המלאי במחסנים, קובץ כח-אדם לחישובי שכר ועוד.

הקבצים גדולים בנפחם וזקוקים לשמירה לפרקי זמן ארוכים, ולכן אין הם יכולים לשכון בזיכרון המחשב. זיכרון המחשב משמש להם "תחנת מעבר" או "לוח טיוטא" בתהליך העיבוד בלבד. הקבצים נשמרים באמצעי אחסון מגנטיים כמו סרט מגנטי, קסטה, תקליטון או תקליט קשיח. לכל מחשב אמצעי אחסון המתאימים לו. כוננני סרטים או דיסקים, או רשמקול קסטות, אשר מחוברים אל המחשב מאפשרים כתיבה מן המחשב, או קריאה אל המחשב, על פי הוראות התכנית.

בפרק זה נכיר דרכים לאחסון נתונים ולניהולם, או כיצד לבצע פעולות קריאה וכתיבה בקבצים. הוראות עיבוד של קבצים הינן הרחבה של שפת בייסיק על ידי יצרני מחשב שונים, אשר התאימו אותן לתכונות ציוד האחסנה והמישקים לחיבור ציוד זה. לכן יש שוני רב יותר בתחום זה בין ניבי בייסיק השונים.

תכנית בייסיק יכולה לטפל בקבצים במבנים שונים: <u>מבנה מחרוזת נתונים</u> (STREAM ORIENTED FILE) <u>ומבנה רשומות</u> (RECORD ORIENTED FILE). בלי להתחשב במבנה הקובץ, נוכל להבחין בכמה שלבי פעולה בתהליך העיבוד:

- הגדרת הקובץ: ציון שם הקובץ.
- פתיחת הקובץ: התחלת העבודה.
- פנייה לקובץ לקריאה או לכתיבה.
- סגירת הקובץ: סיום העבודה.

בכל הפעולות, אשר נעשות בקבצים, יש לציין את שם הקובץ ואת שמות השדות, שמטפלים בהן. בכמה מהניבים אין צורך לפתוח את הקובץ.

מספר דוגמאות להוראות עיבוד של קבצים:

הגדרת קובץ INV	`10 FILES INV`
הגדרת שני קבצים (;מפריד)	`20 FILES INV;TRANS`

קובץ INV (INVENTORY - מלאי) הוא הראשון ברשימה, והתכנית תכיר אותו אח"כ כקובץ מס' 1 (#1). קובץ TRANS (TRANSACTIONS - תנועות) הוא הקובץ השני ברשימה והתכנית תכיר אותו כקובץ מס' 2 (#2). מעתה, כל פנייה לקובץ תהיה באמצעות הציון #1 או#2.

ההוראה

```
30 READ #1,M,K
```

תגרום לקריאת 2 שדות מקובץ #1 (INV) אל המשתנים M ו-K בהתאמה. שים לב שהוראת READ אשר הכרנו קודם שימשה לקריאת נתונים ממאגר DATA, ולא היה לה הסימול של מספר קובץ בראש שורת הפרמטרים.

בשדות M ו-K אפשר להשתמש עתה בהוראות אחרות, כמו B=S+K*3 50

<u>פתיחת קובץ</u> נעשית באמצעות ההוראה OPEN בציון שם הקובץ והפעולה הנדרשת בו: קלט או פלט.

```
5    OPEN #1 IN
10   OPEN #5 OUT
```

בניבים אחרים אין צורך לפתוח קבצים, והתכנית תדע את תפקיד הקובץ על פי ההוראה הראשונה, שהיא תפגוש, הוראת קריאה או הוראת כתיבה. על פי רב דרוש <u>לפתוח קבצי פלט</u> בלבד באמצעות ההוראה SCRATCH, אשר מציבה את הקובץ למצב כתיבה. אין אפשירות לכתוב על קובץ, שלא הוכן לכתיבה.

```
5 FILES INVIN ,INVOUT
10   SCRATCH #2
```

הוראות OPEN או SCRATCH חייבות להופיע בתכנית לפני כל הוראת כתיבה לקובץ. <u>סגירת קובץ</u> גורמת לסיום השימוש בו ולניתוקו באופן לוגי מן התכנית. פעולה זו חשובה במיוחד בקבצי פלט, כדי שהשדות או הרשומות האחרונות יכתבו על הקובץ, ובקובץ יירשם סימן של סוף הקובץ.

```
80   CLOSE #4
```

<u>סימן לסוף קובץ</u> חייב להיות בקובץ כדי לציין לתכנית את המקום, שבו הסתיים רישום הנתונים של הקובץ. אחרי סימן זה אפשר להתחיל בכתיבת נתונים של קובץ חדש, אך גם בעבורו יש לרשום סימן לסוף קובץ. תכנית לניהול הקבצים של מערכת ההפעלה יודעת לרשום סימן זה בעת הוראת CLOSE, ויודעת להבחין בו, כאשר קוראים קובץ, כדי למנוע גלישה לקריאת נתונים, שאינם שייכים לו.

<u>ניהול נתונים בסרטים ובדיסקים</u> נעשה בצורות שונות, ולא נעמוד על כך כאן. בהמשך נציג <u>קבצים סדרתיים</u> (SEQUENTIAL FILES) בלבד.

<u>14.15.1 קובץ במבנה מחרוזת נתונים</u>

בקובץ מחרוזת רשומים הנתונים בזה אחר זה בשדות רצופים, שאורכם לא קבוע. לכן מפריד ביניהם תו רווח, פסיק או סימן אחר מוסכם במערכת. סימן זה "<u>שקוף</u>" למשתמש והוא אינו מכיר אותו ואינו מטפל בו. מערכת התכנה של המחשב כותבת אותו בקובץ לאחר כל שדה ומזהה אותו בעת קריאת הקובץ. המשתמש מטפל בשדות הנתונים שלו <u>בלבד</u>.

1. הוראת קריאה READ

הוראה גורמת לקריאת שדות באופן עוקב בהתאם להוראות READ הפונות לקובץ. מבחינה זו אפשר לראות את קובץ המחרוזת כהרחבה של מאגר DATA, אשר אפשר לכתוב בתוך התכנית.

```
40    READ #2,X,M
```

הוראה זו תגרום לקריאת שני שדות מקובץ 2 ותרשום אותם בזיכרון במשתנים X ו-M בהתאמה.

```
90    READ #2,B,E,G
```

הוראה זו תגרום לקריאת שלושת השדות הבאים בתור בקובץ 2, ותרשום אותם במשתנים G,E,B בהתאמה. כלומר אין ההוראה מתחשבת במבנה הקובץ אם יש בו מבנה רשומות או מבנה של שדות עוקבים. הדבר הוא באחריות המתכנת. על המתכנת להכיר את הנתונים ולקבוע כיצד לקרוא אותם, כדי לגלוש או לא לגלוש מקבוצה או רשומה אחת לשנייה.

תרגיל

יש לקרוא לקובץ תלמידים ולהדפיס את שמות התלמידים, שעושים עבודת גמר במקום בחינת בגרות במקצוע מחשבים. נתון זה נמצא באחד משדות הרשומה.

מבנה הנתונים:

שדה N$ - שם התלמיד
שדות A,B - נתונים שונים
שדה C - קוד=1 אם התלמיד עושה עבודת גמר במחשבים
שדות D,E,F - נתונים שונים

קטע התכנית:

```
10    FILES STUDENTS
20    READ #1,N$,A,B,C,D,E,F
30    IF C#1 GOTO 20     REM IF CODE #1 READ NEXT STUDENT
40    PRINT N$
50    GOTO 20
--- -- -- --
```

10 קריאת כל השדות, כדי שבקריאה הבאה נפנה לקבוצת הנתונים של התלמיד הבא.

30 בדיקת התנאי. שים לב, שלשם הנוחות נעשית בדיקה C#1 ולא C=1.

40 הדפסת שם התלמיד.

50 חזרה לקריאת רשומה.

במקרים מסוימים נוח לקרוא את קבוצת הנתונים בחלקים, בשתי הוראות או יותר, למשל,

```
      - -- -
20    READ #1,N$,A,B,C
      - -- -
      - -- -
60    READ #1,D,E,F
      - -- -
```

לאחר הוראת הקריאה בשורה 60 יש לבצע קריאה לפי הוראה 20 כדי לקבל את השדות המתאימים של קבוצת הנתונים הבאה.

2. הוראת כתיבה PRINT

כתיבת נתונים בקובץ מחרוזת נעשית בהוראת PRINT, לאחר שהקובץ הוכן לכתיבה בהוראת OPEN OUT או SCRATCH OUT. השדות שייכתבו בקובץ יכולים להיות רצופים, אם כותבים בהוראה את הסימן ; (נקודה-פסיק) ביניהם, או שהם יכולים להיכתב לפי כללי העריכה האוטומטיים (בעמודות של 14 תווים, למשל). הוראה זו זהה להוראת ההדפסה שהזכרנו קודם, אלא שכאן נוסף ציון של מספר הקובץ:

```
50    PRINT #1,A,B,C
70    PRINT #2;X;Y;Z
```

ההוראה בשורה 50 תגרום לכתיבת השדות בקובץ 1 בעמודות של 14 תווים. ההוראה בשורה 70 תגרום לכתיבה של השדות בקובץ 2, כשהם צמודים זה לזה, ללא רווח.

כל הוראת PRINT, שאינה מסתיימת בפסיק או בנקודה-פסיק גורמת לכתיבת קבוצת נתונים בקובץ. אם רוצים לכתוב קבוצת נתונים אחת באמצעות מספר הוראות, יש לרשום פסיק או נקודה-פסיק בסוף שורת הנתונים.

14.15.2 קובץ במבנה רשומות

המחשב "אינו יודע" איזה קובץ נכתב או נקרא ואין זה חשוב לו כלל. המחשב כותב וקורא נתונים על פי הוראות התכנית. המבנה של קובץ מחרוזת או קובץ רשומות הוא מבנה לוגי, אשר חשוב למתכנת.

לעתים נוח יותר לפעול בקובץ מחרוזת ולעתים - בקובץ רשומות. רישום הנתונים בקובץ זהה מבחינת המחשב בשתי שיטות העיבוד.

1. הוראת קריאה - INPUT

הוראת הקריאה מקובץ רשומות גורמת לקריאת רשומה אחת והעברת השדות שלה, לפי הסדר, אל רשימת המשתנים.

```
10    FILES POLICE
20    INPUT #1,A$,K,R$,M
```

בהוראה 10 מגדירים קובץ POLICE ובהוראה 20 קוראים ממנו רשומה שבה השדות A$,K,R$ ו-M לפי הסדר. אם מספר השדות ברשומה גדול מ-4, תיקרא הרשומה בשלמותה, אך התכנית תתעלם משאר השדות, ולא תעביר אותם למשתנים כלשהם. אם ברשומה יש פחות שדות מאשר ברשימה של ההוראה בשורה 20, תינתן הודעת שגיאה. כל הוראת קלט תיקרא רשומה חדשה.

2. הוראת קריאה - GET

הוראות הקריאה קוראות את רשומות הקובץ בזו אחר זו. המתכנת לא ירצה לקרוא "מעבר" לקובץ שלו, ולשם כך עליו לבדוק, אם הגיע לסוף הקובץ. הוא יכול לעשות זאת באמצעות כתיבת רשומה אחרונה עם קוד מיוחד, או להשאיר למחשב את "הטרחה" לזהות את סימן סוף הקובץ, שנרשם בעת הוראת CLOSE.

```
25    GET #3,A,B,EOF 80
```

ההוראה 25 תגרום לקריאת שני שדות A,B מקובץ הקלט 3 בהתאם למבנה הרשומה. כאשר התכנית תזהה סוף קובץ (EOF - END OF FILE), היא תדלג לשורה 80.

3. הוראת כתיבה - WRITE

כתיבת רשומות בקובץ רשומות נעשית באמצעות ההוראה WRITE. כל הוראה גורמת לכתיבת רשומת פלט אחת בקובץ, שמספרו רשום בהוראה. יש להכין את הקובץ לכתיבה בהוראת OPEN OUT או SCRATCH.

כתיבה לפי עמודות סטנדרטיות	`40    WRITE #1,A,B`
כתיבה של שדות צמודים	`50    WRITE #2,R;N`

אם יש לכתוב רשומה אחת באמצעות מספר הוראות, יש לכתוב פסיק או נקודה-פסיק בסוף רשימת המשתנים. המשתנים ייכתבו

ברשומה צמודים, אם נרשם ביניהם הסימן נקודה-פסיק, והס-ייכתבו לפי עמודות העריכה הסטנדרטיות, אם ייכתב ביניהם הסימן פסיק.

4. הוראת כתיבה WRITE USING

הוראת כתיבה זו משתמשת בשורת תבנית לעריכת הנתונים לפלט, בדומה להוראה PRINT USING, אשר שימשה להדפסה ולתצוגה של נתונים. כללי העריכה שווים.

```
-- -- --
 70   WRITE #2, USING 20,A,B
-- -- --
120   :## ##.#
```

הערכים מתוך השדות A ו-B ייכתבו בהתאם לכללי העריכה, שנקבעו בשורת התבנית.

דוגמאות :

A , B	נתונים ערוכים לכתיבה ##	##.#
53, 124	53	12.4
82.5, 200	82	20.0
4573, 7	573	.7

14.15.3 הוראות בדיקה לסיום קובץ

ברוב הניבים של בייסיק קיימות שתי הוראות, אשר מקילות על המתכנת את הבדיקה אם הגיע לסופו של קובץ הקלט. הבדיקה נעשית על פי הסימן של סוף הקובץ, אשר נרשם בעת ביצוע הוראת CLOSE, ואינו מחייב את המתכנת לכתוב רשומת סיום בעלת קוד מיוחד.

1. האם סוף הקובץ - IF END

הוראה זו בודקת את סימן סוף הקובץ. אם הקובץ הסתיים, מדלגת התכנית לשורה, שמספרה רשום בהוראה, אם לא - היא ממשיכה בשורה העוקבת.

```
20   IF END #3 GOTO 70
```

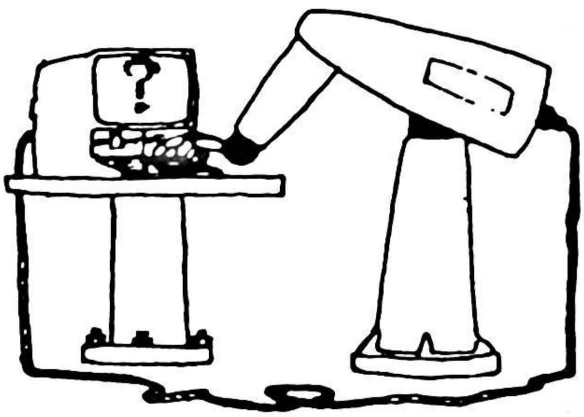
אפשר לכתוב THEN במקום GOTO, כבשאר הוראות הדילוג.

```
10    FILES CARS
15    N=0
20    IF END #1 GOTO 70
30    INPUT #1,M$,E$,B
40    PRINT M$,E$      REM PRINT MODEL AND YEAR
50    N=N+1            REM COUNT NO. OF CARS
60    GOTO 20
70    PRINT N          REM PRINT TOTAL COUNT
80    END
```

2. האם יש נתונים נוספים - IF MORE

הוראה זו בודקת, אם יש נתונים נוספים לקריאה. אם יש נתונים נוספים, מדלגת התכנית לשורה, שמספרה רשום בהוראה, ואם לא - היא ממשיכה בשורה העוקבת.

```
10    FILES CARS
15    N=0
20    INPUT #1,M$,E$,B
30    PRINT M$,E$      REM PRINT MODEL AND YEAR
40    N=N+1            REM COUNT NO. OF CARS
50    IF MORE #1 GOTO 20
60    PRINT N          REM PRINT TOTAL COUNT
70    END
```



"שיתוף פעולה"

ביבליוגרפיה

הרשימה הביבליוגרפית כוללת מגוון של ספרים, חוברות ותקופונים בנושאים שנדונו בספר זה. מהיותו ספר מבוא כללי להכרת המחשב האלקטרוני ונושאי עיבוד הנתונים, כוללת גם הרשימה שלהלן מגוון של ספרות בנושאים שונים. לנוחות הקורא מוצג החומר לפי הנושאים.

בחרנו להציג רק חומר ביבליוגרפי הקיים בעברית, בידיעה שרוב הקוראים עושים את צעדיהם הראשונים בתחום המופלא של עיבוד הנתונים, והם יעדיפו את הקריאה בעברית.

בנוסף, קיימת ספרות עניפה באנגלית, והקורא המעוניין יכול להשיגה. יש לציין את הפרסומים של יצרני מחשבים ובתי-תוכנה, אשר מציגים מערכות חמרה ותכנה, אשר הם משווקים. חלק מהפרסומים, אשר מוגדרים כ-General Information Manual, מתאימים גם לקורא המתחיל.

1. <u>מבוא למחשבים</u>

א. בעל שם יעקב
<u>יסודות תורת המחשבים</u>, המכון לאמצעי הוראה, 1970

ב. גולדנברג אליעזר
<u>מבוא למחשבים</u>, מכלול, 1975

ג. מצפי סמי
<u>הקלט למחשב</u>, הוד-עמי, 1978

ד. עמיהוד יצחק
<u>המחשב האלקטרוני ועיבוד הנתונים</u>, הוד-עמי, 1968

ה. - <u>יסודות תורת המחשב</u> (2 חלקים), מט"ח, 1979

ו. - <u>מבוא לתכנות ולהכרת המחשב</u> (4 חלקים), האוניברסיטה הפתוחה, 1978

2. תכנות, תרשימי זרימה, טבלאות החלטה

א. מתיא פנחס
טבלאות אמת והחלטה, המכון לפריון העבודה והייצור, 1983

ב. ניר אי

שפת בייסיק הלכה למעשה, א.ר.ט שרותים לאופסט, 1982

ג. סטשבסקי, זילברמן
תרשימי זרימה ושפת התכנות בייסיק, מפעלים אוניברסיטאיים, 1978.

ד. עמיהוד יצחק
תכנון ותכנות יישומי מחשבים (תרשימי זרימה, פורטרן, קובול, PL/1), הוד-עמי, 1975

3. מערכת מידע ניהולי וניתוח מערכות

א. בורוביץ ישראל, נוימן זאב, מלמד יוסף
מערכות מידע ממוחשבות, ציריקובר, 1984

ב. הדס מיכאל
מערכות מידע, המכון לפריון העבודה והייצור, 1975

ג. מתיוס רון
מערכות מידע ניהולי, סדן והמכון לפריון העבודה והייצור, 1973.

ד. נוימן זאב, בורוביץ ישראל
מחשבים ומערכות מידע, גומא, 1976

ה. עמיהוד יצחק, קורפל יונתן
יסודות בניתוח ותכנון מערכות מידע, הוד-עמי, 1980

ו. קורפל יונתן
ארועים בניתוח מערכות ענ"א, תרגילים בניתוח מערכות ענ"א, המכון לפריון העבודה, 1980.

ז. תלם משה, אולשנסקי אביגדור
המשרד הממוחשב ציריקובר, 1985

4. ארגון קבצים, בסיסי נתונים

א. הייפרמן רז
ארגון נתונים וניהול קבצים, הוד-עמי, 1982

ב. הייפרמן רז
בסיסי נתונים - עקרונות, מודלים ויישומים, הוד-עמי, 1984

5. תקשורת

א. סקופ אריה
מערכות תקשורת נתונים, הוד-עמי, 1981

ב. רוט מאיר
עקרונות של מערכות תקשורת, יבמ, 1983

6. מיקרו-מחשבים

א. הראל אהרון
מיקרו-מחשבים וה-6800, חמרה ותכנה, הוד-עמי, 1983

ב. חיים שפירא
מיקרו-מחשב, אלי, 1981

ג. אריה נחום
יסודות המיקרו-מחשב 6800, 8085 - לימוד ותרגול, אס.סי.אס.

ד. - מיקרופססור תכנה וחמרה - מט"ח, 1982

7. בטיחות נתונים ומידע

א. בר-חוה נורי, תלם משה
מחשבים וסיכונים, ציריקובר, 1983

ב. יובל שלמה
עבריינות במחשבים, 1977

8. <u>תיעוד מערכות</u>

א. - <u>נהלי תיעוד ליחידות ענ"א</u>, המכון לפריון העבודה והייצור, 1980

ב. קורפל יונתן
<u>תיעוד לניתוח ותכנון מערכות ממוחשבות</u>, המכון לפריון העבודה והייצור, 1980

9. <u>כתבי-עת</u>

א. <u>אנשים ומחשבים</u>,

ב. <u>האלקטרוניקה בישראל</u>, ס.ג.מ.

ג. <u>המחשב האישי</u>,

ד. <u>מחשבים</u>, מירב בע"מ

ה. <u>מידעון</u>, איגוד מנתחי מערכות ע"י המכון לפריון העבודה והייצור

ו. <u>מעשה חושב</u>, איל"א

10. <u>שונות</u>

א. פרידמן אלן
<u>לכסיקון המחשב</u>, דליה פלד מוציאים לאור, 1984

ב. מיילר דונה
<u>עצור לפני שתקנה מחשב</u>, דליה פלד מוציאים לאור, 1984

אינדקס ומילון מונחים

מילון המונחים הקצר המצורף מגיש לקורא את המונחים העיקריים שהוא נזקק להם כאשר הוא מתחיל בלימודי המחשב. ההצגה של המונחים בעברית ובאנגלית תסייע לקורא בעת עיון בספרות העזר העניפה.

במילון משולב גם אינדקס, אשר נותן את מראי המקום העיקריים של הדיון בנושא או במושג.

סדר המיון הוא כפול, בהתחלה לפי התמליל בעברית ואחרי כן לפי המושגים באנגלית.

- מהו מחשב ואיך הוא פועל?
- איך משתלב המחשב בעבודת המנהל ואיש המקצוע?
- איך מתכננים עבודה למחשב, ואיך אומרים לו מה לעשות?
- איך שומר המחשב את הנתונים ומאפשר לנו לאחזר אותם?
- מהי תקשורת נתונים ואיך ניתן שרות במסופים?
- מהם מיני-מחשב, מיקרו-מחשב ומחשב אישי?
- איך מפעילים שרותי מידע בארגון?
- איך מתכנתים בשפת בייסיק (BASIC)?

המחשב האלקטרוני - עיבוד נתונים ותכנות בייסיק מציג בפני הקורא והלומד תמונה מקיפה של מערכות החומרה והתוכנה של מחשבים מבלי להתייחס למערכת מסויימת. קשת הנושאים רחבה ומאפשרת לקורא להתוודע אל תחומים שונים בעיבוד נתונים אוטומטי. שפע השרטוטים, האיורים והתמונות משלימים את הכתוב. פרק מקיף **בתכנות בייסיק** מיועד לכל מי שרוצה להכיר את השפה וללמוד לתכנת בה.

הספר ערוך לפי הדרישות של תוכנית הלימודים החדשה של משרד החינוך והתרבות עבור מגמות מחשבים - ענ"א בבתי הספר העל-יסודיים ובמכללות להנדסאים, לפי דרישות מרכז ענ"א במכון לפריון העבודה והייצור, ולפי תוכניות הלימודים במכללות למחשבים.

הספר נועד לכל מי שמעוניין להכיר וללמוד מהו מחשב, ואיך אפשר להשתמש גם במערכי עיבוד נתונים. מנהלים, מהנדסים ומכנאים, סטודנטים ותלמידים במגמות מחשבים ובמכללות ללמודי המחשב ימצאו בספר את מבוקשם.

המחברים

יחיאל האן, יליד 1932, בוגר האוניברסיטה העברית בירושלים בכלכלה וסוציולוגיה. לאחר 10 שנים של הוראה ומינהל חינוכי הצטרף ב-1969 לחברת יבם. מילא תפקידים מגוונים בהדרכה ובהנדסת מערכות. מנהל את תכנית ההכשרה לעובדים החדשים באגף השיווק ביבם. מזה 15 שנה מרצה במקצועות ענ"א במוסדות שונים להשכלה גבוהה.

יצחק עמיהוד, יליד 193?, MSC בהנדסת תעשיה וניהול מטעם הטכניון בחיפה. עוסק בנושאי עיבוד נתונים אוטומטי מאז 1959, תחילה בצה"ל ומשנת 1963 - בחברת יבם. מילא קשת רחבה של תפקידים בתחום שונים והנדסת ...
המקצועות ענ"א ...
כיו"ר. ...